AD-A047 256 SYSTEM SCIENCES INC BETHESDA MD F/G 15/7
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM. (QUICK). USE--ETC(U)
JUN 77 D J SANDERS, P F MAYKRANTZ, J M HERRIN DCA100-75-C-0019
UNCLASSIFIED CCTC-CSM-UM-9-77-VOL-1 NL

1 OF 2
ADA
047256

C
C
T
C

12
mc

COMPUTER SYSTEM MANUAL
CSM UM 9-77
VOLUME I
1 JUNE 1977

**COMMAND
& CONTROL
TECHNICAL
CENTER**

THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK)

Volume I

Data Management Subsystem

USERS MANUAL

DEFENSE
COMMUNICATIONS
AGENCY

COMMAND AND CONTROL TECHNICAL CENTER

18 CCTC 19

Computer System Manual CSM-UM-9-77 - Vol-1

11 1 June 1977 12 159p.

6

THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK).

Users Manual.

Volume I. Data Management Subsystem.

2-A046 088

SUBMITTED BY:

C G Thompson

C. G. THOMPSON
Project Officer

15 DCA100-75-C-0019

APPROVED BY:

R E Harshbarger

R. E. HARSHBARGER
Acting Deputy Director
NMCS ADP

10 Dale J. /Sanders,
Paul F. M. /Maykrantz,
Jim M. /Herrin
Edward F. /Bersson

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.

Approved for public release; distribution unlimited.

388286

mt

ACKNOWLEDGMENT

This documentation was prepared under the direction of the Chief for Military Studies and Analysis, CCTC, in response to a requirement of the Studies, Analysis, and Gaming Agency, Organization of the Joint Chiefs of Staff. Technical support was provided by System Sciences, Incorporated under Contract Number DCA100-75-C-0019.

CONTENTS

Section Page

Appendix

FIGURES

TABLES

ABSTRACT

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide output summaries, and produce tapes to simulator subsystems external to QUICK. QUICK has been programmed in FORTRAN for use on the CCTC HIS 6080 computer system.

The QUICK Users Manual consists of four volumes: Volume I, Data Management Subsystem; Volume II, Weapon/Target Identification Subsystem; Volume III, Weapon Allocation Subsystem, Volume IV, Sortie Generation Subsystem. The Users Manual complements the other QUICK Computer System Manuals to facilitate application of the war gaming system. This volume, Volume I, provides instructions for using the Data Management Subsystem. It is intended for the CCTC user/analyst who is concerned with preparing the data base for a war game, selecting optional features of QUICK, designating control parameters, submitting computer jobs, and analyzing computer output. Companion documents are:

a. PROGRAM MAINTENANCE MANUAL
Computer System Manual CSM MM 9-77, Volume I
Computer System Manual CSM MM 9-77, Volume II
Computer System Manual CSM MM 9-74, Volume III
Computer System Manual CSM MM 9-74, Volume IV
Provides detailed instructions for maintenance of the system

b. TECHNICAL MEMORANDUM
Technical Memorandum TM 153-77
Provides a nontechnical description of the system for senior management personnel

SECTION 1. GENERAL

1.1 Purpose

This manual is intended to inform the CCTC user/analyst on how to prepare control cards; structure execution (run) decks; prepare computer job requests; and understand the associated computer output, to include the recognition of error messages for the Data Management subsystem of QUICK. It complements information contained in the Program Maintenance Manuals on the QUICK system. The abstract of this document references other documents describing QUICK.

1.2 Program Environment - QUICK System Overview

The QUICK-Reacting General War Gaming System (QUICK) is a unique analytical tool which provides a comprehensiveness to strategic war gaming that has not been available through other computerized models. QUICK is designed to assist in the study of strategic conflicts involving a large-scale exchange of nuclear weapons. Toward this end, the system encompasses three major capabilities which are applicable to a wide range of studies: first, for a given offensive missile and bomber force and a specific set of targets, QUICK produces a detailed plan of attack which is near optimum for the conditions specified by the user. Second, it provides an expected-value estimate of the results of that attack. Finally, it produces input tapes to simulator subsystems external to QUICK.

QUICK is structured into four major subsystems: Data Management, Weapon/Target Identification, Weapon Allocation, and Sortie Generation. The principal tasks associated with each of these functional subsystems are summarized below.

a. Data Management: Assembles and reformats the target and non-target data required for a particular plan

b. Weapon/Target Identification: Selects and processes the Red and/or Blue Forces prespecified for a particular plan

c. Weapon Allocation: Allocates offensive weapons to selected targets

d. Sortie Generation: Prepares and evaluates missile and bomber attack plans.

Figure 1 displays the modules which comprise each subsystem.

Figure 2 illustrates the communication with the Central Operations Processor (COP) or executive software and the entire procedural and informational flow within the QUICK system. The communication lines infer

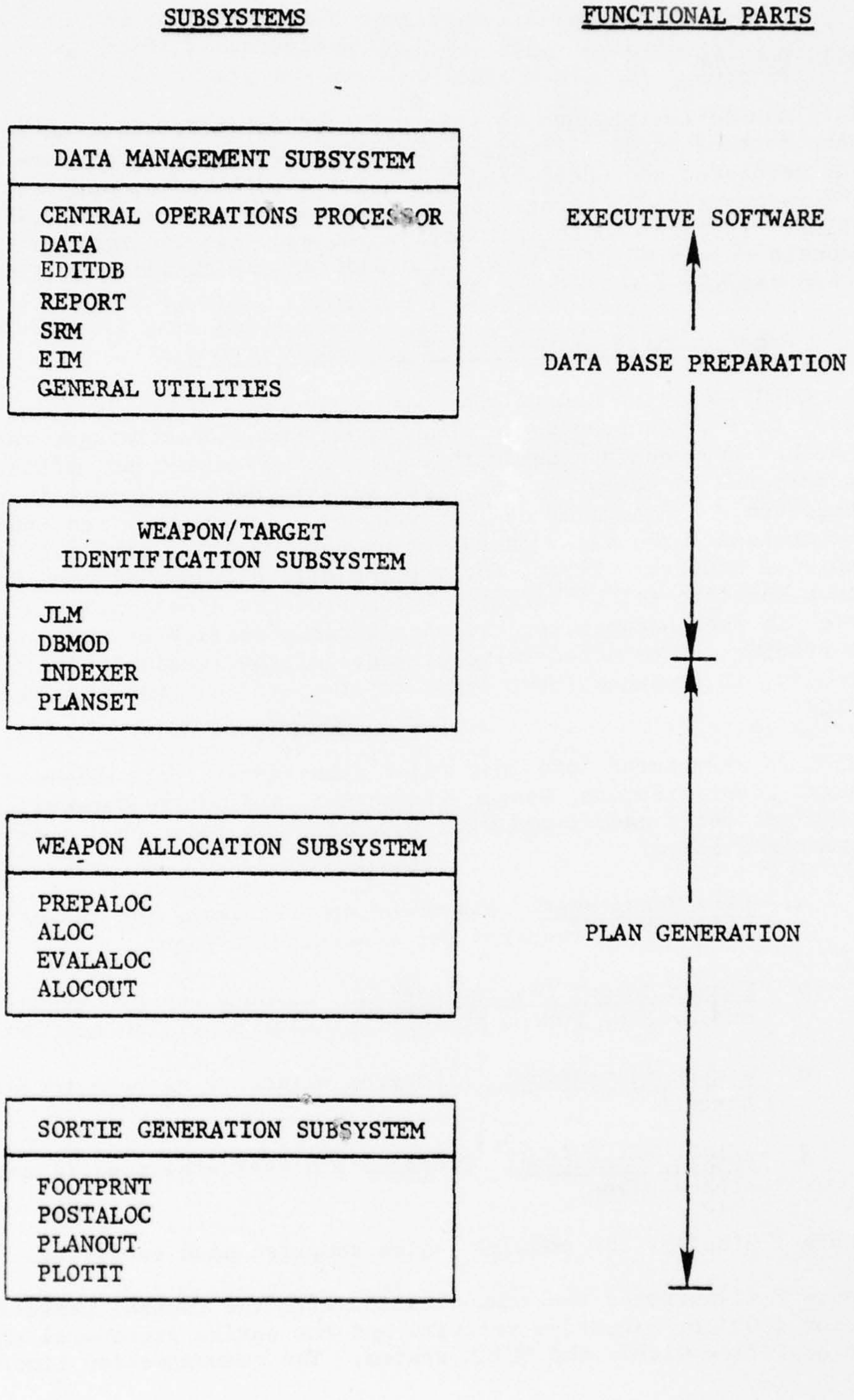


Figure 1. Major Subsystems of the QUICK System

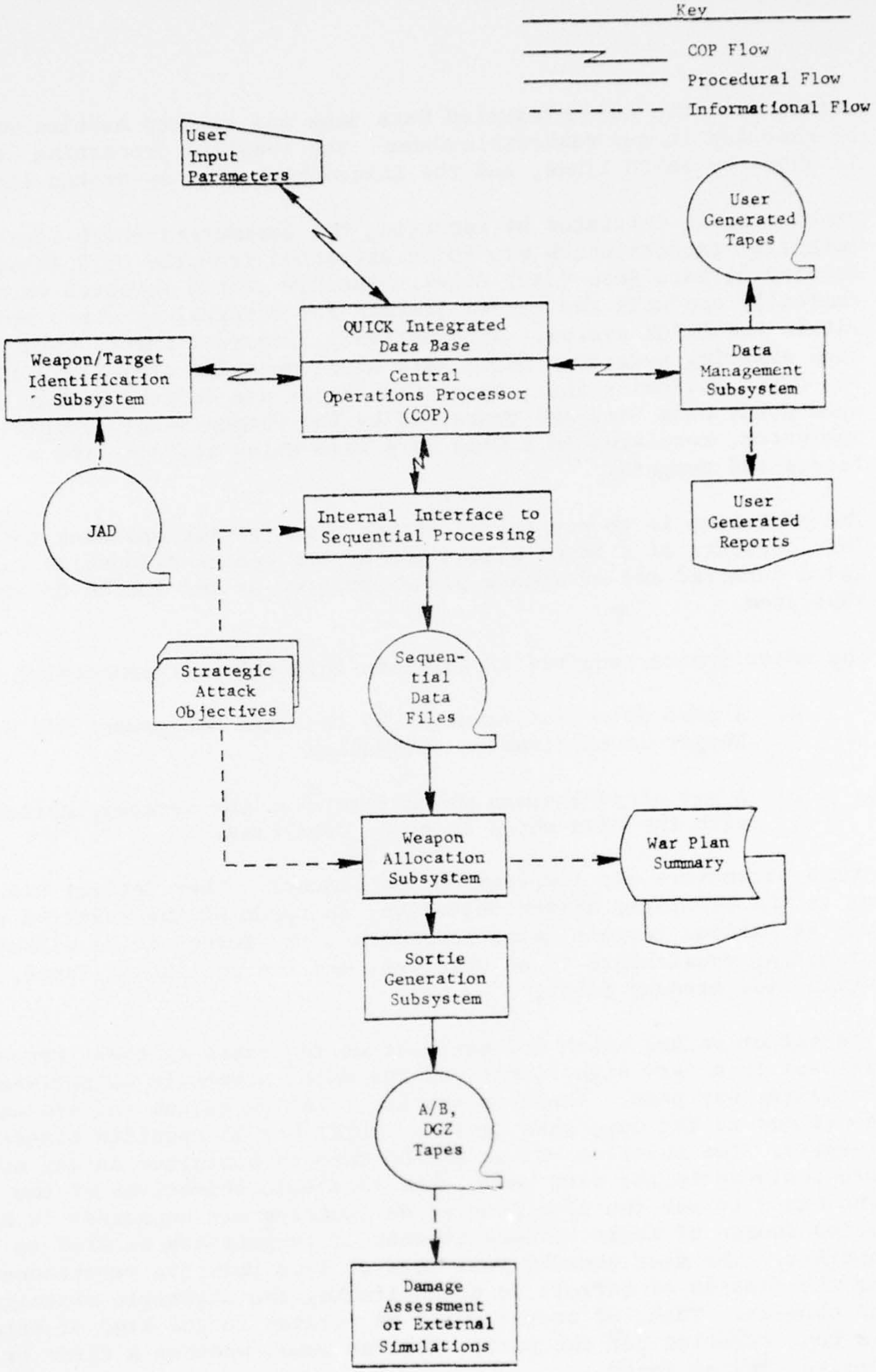

Key
COP Flow
Procedural Flow
Informational Flow
User Input Parameters
User Generated Tapes
QUICK Integrated Data Base
Central Operations Processor (COP)
Weapon/Target Identification Subsystem
Data Management Subsystem
JAD
Internal Interface to Sequential Processing
User Generated Reports
Strategic Attack Objectives
Sequential Data Files
Weapon Allocation Subsystem
War Plan Summary
Sortie Generation Subsystem
A/B, DGZ Tapes
Damage Assessment or External Simulations

Figure 2. Procedure and Information FLOW in QUICK/HIS 6000

action with the COP Integrated Data Base and related modules which may be executed in any reasonable order. The required processing sequence is shown by solid lines, and the information flow by broken lines.

Processing is initiated by inputting the parameters which identify the potential targets which are to be extracted from the CCTC Joint Resource Assessment Data Base (JAD) files. The COP stores selected data and dynamically conducts the proper linkage for referral by other modules within the QUICK system. Alternatively, required target data is obtained from existing updatable QUICK Data Bases and also stored and linked by the COP. Following this, specified forces are defined within the developed QUICK Data Base and processed by the Weapon/Target Identification subsystem, resulting in a Game Data Base which reflects the selected forces and targets.

The next step is to prepare an attack plan for the opposing forces. This consists of a force allocation by the Weapon Allocation subsystem, and a detailed set of attack plans prepared by the Sortie Generation subsystem.

The major inputs required to initiate this phase of processing are:

a. A game data base prepared by the Data Management and Weapon/ Target Identification subsystems

b. A set of parameters which relate to the strategy associated with the plan which is to be developed.

These parameters are supplied by the planner. They reflect his views as to the strategic attack objective, in terms of the relative values of the various targets being considered, the forces to be withheld, the targeting constraints to be observed, and the initiating force, i.e., which side attacks first.

The target values which are computed on the basis of these parameters reflect in a very significant way the major strategic objectives of the resultant war plan. These values are relative values and are partially contained in the data base itself. QUICK has 15 specific classes of targets. The relative values of the targets contained in any one class are included in the data base: the strategic objectives of the planner who wants to use the plan generation function are expressed in how the value scales of these various classes of targets are related to one another. The user thereby puts more or less relative importance on each of the classes of targets in accomplishing the strategic objectives that he chooses. This, of course, will be related to the kind of strategy he is contemplating for the particular war game, whether a first or second strike, and so forth.

Having established a value for each target, the plan generation phase mathematically allocates the weapons (e.g., Red weapons to Blue targets) and prepares the detailed missile and bomber attack plans. If desired, the plans may be printed, inspected, and altered by changing the attack objectives and repeating the process. The series of Red missile and bomber events corresponding to the sortie plan is prepared in a form suitable for input to external simulators. As a user option, a war plan summary is provided which includes an expected-value estimate of the results of the attack. In addition, the aim point, Desired Ground Zero (DGZ), for each planned weapon can be output for subsequent evaluation utilizing an external damage assessment system.

While the system can proceed automatically through all steps if desired, it may be halted at the end of each module (or program), and the available output inspected for correctness and adequacy. There are standard outputs from each module and, also, mechanisms exist for producing generalized outputs (either tape/disk or printed reports). Through the COP, users may request output in any format of data items that reside within the Integrated Data Base. The user has complete flexibility in displaying needed reports.

1.3 Data Management Subsystem Overview

The Data Management subsystem consists of the executive software (called COP) which maintains overall control of the entire system plus modules necessary to create, change, maintain, or display the QUICK data base. Also included is a utility package (sometimes called library) which consists of subroutines and functions which perform a variety of support tasks common to several system programs. General descriptions of these functions and their organization in subsequent sections of this volume are provided in the following subsections.

1.3.1 QUICK Integrated Data Base. QUICK employs the HIS Integrated Data Store (IDS) system to maintain and control the data base. This type of data base has all record types interconnected in some logical fashion such that data items may be queried in a generalized manner.

1.3.2 QUICK Generalized Text English Commands. The user directs the execution of QUICK through text English commands which are imperative sentences that provide meaning to the executive software. These commands permit data construction, access, maintenance, validation and display, as well as provide QUICK module executions.

1.3.3 QUICK System Central Operations Processor (COP). All authority for directing the QUICK system resides within the COP. This includes all interfaces with the system and the user, as well as all flow between the data base and related modules. COP also controls the execution of all modules.

1.3.4 Data Management Modules. The Data Management modules consist of modules DATA, EDIT, REPORT, SRM, and EIM. These modules and supporting subroutines create, maintain, and display the data base. Their order of execution is not rigid. A sequence of execution holds only for the obvious situations such as a display request is possible only after the requested display items have been created and stored within the data base.

1.3.5 General Utilities. The general utilities consist of subroutines, and functions which perform a variety of tasks throughout the QUICK system. Among the tasks performed by the general utilities are, for example, error diagnostics, supporting positional and distance calculation functions, tallying of targets, etc. Because the user does not interface with these subroutines, they are not further discussed in this manual.

1.4 Computer Software Environment and Programming Language

The QUICK system runs under control of the HIS 6000 GCOS (General Comprehensive Operating Supervisor). Except for a few utility type subroutines which are written in the Generalized Macro Assembler Program (GMAP) language of the HIS 6000 computer system, all QUICK programs are programmed in the FORmula TRANslator (FORTRAN) computer language.

FORTRAN is the coding language. The data base is created and maintained using the HIS Integrated Data Store (IDS) subsystem. All data items necessary for a complete QUICK system execution reside within the IDS structure.

Also, in order to easily generate Job Control Language (JCL) instructions and to provide for automatic updates to master source files, a system as outlined in Appendix F is provided.

1.5 Equipment Environment

The QUICK system is operational on the CCTC HIS 6000 computer system. Available to the QUICK system in addition to standard peripheral equipment are three 6000 processing modules, four 6000 system controllers, six 64K memory modules, and two seven-channel and 14 nine-channel magnetic tape handlers. Type 181 and 191 disk storage units are available for permanent files.

Types of available remote access devices relevant to the interactive capability are:

a. VIP 786W - CRT Subsystem (EIA or 188C)
b. KSR 33B - Teletype 188C (TTY)
c. RLP 300 - Remote Printer (EIA or 188C)
d. IBM 2741 - Communications terminal
e. VIP 7705 - CRT Subsystem (EIA or 188C)

1.6 Limits of the QUICK System

The QUICK data base, in itself, is limited only by the capacity of the storage area. However, it is prudent to limit certain data items in order to maintain reasonable limits (compute time and core) during execution. Also, by nature of employed algorithms, certain mathematical calculations could possibly not function properly if totally open-ended parameters were permitted. Therefore, limits as given in table 1 are used within the QUICK system. Note that no restrictions are placed on the number of targets.

1.7 Organization of Users Manual, Volume I

In general, each major section of this manual is subdivided into two major subsections. These are:

a. Module input - details the set-up of input data files and how they are used in a given module

b. Module output - details the scope and content of module output, with noted examples.

Note that generally most computer activities refer to executions in terms of programs. The compatible computer program within the QUICK system is the COP. Modules, or a set of subroutines necessary to perform some function, are executed by the COP.

Table 1. Limits of the QUICK System

TARGET CLASS DATA	MAXIMUM NUMBER
Targets per plan	Open-ended
Target Classes/side	15
Targets (Target Class Items)	Open-ended
Target Complexes	Open-ended
Target Elements per Complex	40
Targets Defended by Terminal Antiballistic Missile Interceptors	Open-ended
Target Vulnerabilities (Number of unique entries)	255
SUPPORTING CLASS DATA	
Warhead Types	50
Payload Types	40
Bombs plus ASMs per payload	10
Air-to-Surface Missile (ASM) Types	20
Command/Control Regions	20
Corridors (Penetration)	30
Depenetration Corridors	50
Recovery Bases (Bomber) per Depenetration Point	4
Refuel Points (User-Directed)	20
Weapon Systems	100*
OTHER CONSTRAINTS	
Allocation of Weapons per Target Without Terminal BMD	30
Target with Terminal BMD	30**
Fixed (User-Directed) Weapon Assignments	Open-ended
Flags (Allocation Constraint Indicator)	9
Weapon Groups	
Number of Groups	250
Missile/Bomber per Group	150
Weapons per Group	1000
Tanker Bases	60

[*] Represents sum total of missiles plus bombers. Any mix within the stated upper bound is permitted.

[**] Weapons from a total of 30 weapon groups may be assigned with no limit on the maximum number of weapons.

SECTION 2. THE QUICK INTEGRATED DATA BASE

2.1 General

The basic concepts relating to the composition and construction of the QUICK data base are presented in Volume I of the Program Maintenance Manual. The information contained in this User's volume provides the user/analyst with additional instructions related to the techniques, requirements and procedures for the development and/or update of a QUICK data base. While the scope of this Users Manual does not include a detailed treatment of the QUICK data base generation developed by CCTC, it does provide the user/analyst with the background and knowledge necessary to prepare inputs required to create a basic data base for subsequent processing by the Data Management subsystem.

This section pertains only to suggestions of war gaming attribute organizations and creation. No mention is made as to the exact techniques of inputting these data items into the QUICK system. The user/system interface language is developed within section 3 of this manual.

2.2 Data Base Entries

2.2.1 General. Any data file has some entry through which data is accessed. For standard sequential data files, the point of entry (or access) is simply the position of the first physical record within the file. The powerful generalization of the QUICK IDS data base file is the existence of multiple entries. Through knowledge and understanding of these entries the user/analyst can efficiently control desired actions.

The QUICK IDS data base is organized into collections of data records called CLASS (a war gaming attribute). A CLASS, then, owns multiple records where each record contains similar data items or attributes. CLASS values are considered to be either target classes or non-target classes.

With the cited data base organization, the most efficient entry is through a specification of the attribute CLASS. (Since many sides may be stored simultaneously, the attribute SIDE should also be set.) General entries for target and non-target data and, also, entries where CLASS is not specified are discussed below.

2.2.2 Target Classes. A target is defined in the most general sense as being any geographic point that could be targeted. Hence, offensive weapon launch bases according to this definition are included under the heading of a target class.

There are up to 15 classes per side reserved for storing target data. These classes are created by setting attributes CLASS and SIDE to unique values. The target class name is any user input with a few reservations for special control names, for in order to automatically calculate needed parameters from user inputs, the implemented code within the QUICK system searches for the existence of several special class names. These class names are outlined in table 2. For example, if the coded calculations for urban/industrial targets are to be executed, they must be defined under CLASS = U/I.

2.2.3 Non-target Classes. All other war gaming related data not entered as targets are collectively defined as being non-target classes and, also, are organized according to attribute value pairs of CLASS and SIDE (see table 3). These classes mainly provide for the organization of weapon characteristics, payload and warhead tables, and geographic related data. The linkage is somewhat detailed and fully outlined in appendixes of this manual.

2.2.4 Entries Other Than CLASS. As cited, the easiest (hence fastest) access to data items is the specification of attribute CLASS. However, entries to any data item within the data base are possible; some desirable, some not.

Most classes are subdivided according to attribute TYPE; such as TYPE= B-52 under CLASS=BOMBER. Therefore, TYPE is another level of data base organization and is recommended for use in data acquisition. Setting TYPE by itself is permissible, but setting both TYPE and CLASS is more powerful.

A single target record is readily retrieved through the attribute DESIG.

Generally, the more information supplied, the easier it is to acquire the requested record. An example of the worst case would be to request a record where attribute NAME equals some input. In order to find the needed NAME, the operating system would have to search all records wherein the attribute NAME resides; obviously a lengthy (but possible) process. However, if the system is supplied more attributes, the search scheme is eased. For instance, a request to find NAME under a certain TYPE is not a difficult request to honor.

2.3 Data Base Preparation

A complete and detailed explanation of which attributes must be defined is outlined within appendixes of this manual. Appendixes associated with data base preparation are:

Table 2. Reserved Target Class Names

CLASS MNEMONIC	DATA CATEGORY
MISSIL	Offensive missiles
BOMBER	Offensive bombers
TANKER	Tankers
NAVAL	Naval targets
U/I	Urban/industrial targets

Table 3. Non-Target Gaming Classes

CLASS MNEMONIC	PURPOSE
WEPGRP	Contains weapon group data
PENCOR DEPCOR	Defines penetration and depenetration corridors
COMPLX	Contains elements of targets defined as being in a complex (formed from target classes)
REFUEL	Defines refuel points
REGION	Defines regions
PAYLOD	Identifies weapons and counts carried by a missile or bomber
BOMB ASM RV MRV MIRV FACTOR	Provides warhead characteristics for gravity bombs, ASMs, single reentry vehicles (RV) and other types
MSLWEP BMBWEP TNKWEP	Contains weapon system characteristics for missiles, bombers and tankers
RECOV	Provides special entry to recovery bases

- Appendix A - Attribute definition
- Appendix B - Directory definition (maximum, minimum, and default values for all attributes)
- Appendix C - Attribute Name listed in numeric order
- Appendix D - Non Attribute Word Identification Number
- Appendix E - Comments on the order and nature of inputting data items
- Appendix F - User procedures for executing the QUICK System

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 3. QUICKS GENERALIZED TEXT ENGLISH COMMAND LANGUAGE

3.1 General

The Central Operations Processor (COP), or QUICKs executive module, acts as an intermediary between the user and the operation system, and provides the communication between the QUICK system and the integrated data base. The COP acts accordingly by interpreting user commands written in imperative text English formats. These commands permit data construction, access, maintenance, validation and display, as well as provide QUICK module execution.

QUICKs text English command language contains lists of permissible entries (or words) that have meaning to the COP. In addition, the entries have a structure, or syntax, whereby separate entries (words) are strung together in a defined, logical fashion to provide mutual user understanding. QUICKs text English command language, then, is similar to formal English, which contains definable words and a grammar or set of rules describing how these words may be presented. As with formal English, words for QUICKs command language (and syntax) are grouped into sets from which the grammar is defined. Specifically, words are defined (grammatically) as belonging to groups of Verbs, Adverbs, Special words, Operators, Nulls and Attributes. Definitions of these word groupings are:

- o Attributes - List of variables necessary for data base construction (see appendix A)
- o Verbs - A list of words describing the <u>Action</u> of each sentence
- o Adverbs - A list of words used to modify the verb or introduce "Clauses"
- o Special Words - A list of words that cannot be classified as attributes, verbs, or adverbs
- o Operators - A list of words or symbols necessary for defining expressions
- o Nulls - A list of words that make a sentence readable but are not necessary for understanding a command
- o Syntax Directory - Defines how words within a sentence may be connected

3.1.1 Sentences. Each command generated by the user for input to the QUICK system is written as an imperative sentence. Since the system interprets all sentences, COP is the assumed subject. Some characteristics of sentences are listed below:

- A verb initiates all sentences;
- A sentence may contain only a verb;
- Sentences are interpreted left to right;
- Sentences may be of any length;
- Blank(s) separates words;
- More than one sentence may be placed on a record (either a data card or a VIP transmittal line);
- Sentences may begin in any column of the record (card or VIP).

From the verb, the system has the ability to ascertain and execute the correct computer module and provide as input any text that follows the verb up to the occurrence of the next verb, which initiates a new set of instructions. The appearance of a following verb does not imply the initiation of a new module. A continuous list of sentences may be provided to a given module. For some cases, data records may be changed, new ones created and/or old ones deleted; all of these commands are met within one module, but separate verbs are required to define objectives.

3.1.2 Format Conventions. Sentences are not rigid; flexibility is permitted as to what may or may not be included. Therefore, conventions such as those shown in table 4 shall be adhered to for the discussions of command construction. Later sections within this manual will outline commands necessary for each module execution and the input descriptions will be discussed in terms of the conventions given in table 4. Most of the symbols are self-explanatory, however, an example of a command is given below to clarify any uncertainty. Consider the command:

$$\underline{\text{CHANGE}}\ \underline{\text{WHERE}}\ \underline{\text{attribute}}\ \begin{Bmatrix}\underline{\text{EQUALS}}\\ \underline{=}\end{Bmatrix}\ \underline{\text{value}}$$

$$\underline{\text{SETTING}}\ \underline{\text{YIELD}}\ \begin{Bmatrix}\underline{\text{EQUAL}}\\ \underline{=}\end{Bmatrix}\ \text{TO}\ \underline{4}$$

With the exception of the word 'TO' all words given in the example are necessary for proper interpretation. In two cases, the word 'EQUAL' or the symbol '=' provides alternative methods of requesting identical objectives; either method may be used. The lower case entries imply that any word contained within the attribute list (SIDE, CLASS, for example) can be set to given values (BLUE, U/I, etc.). The command,

Table 4. Sentence Format Conventions

SYMBOLS	DESCRIPTION
[ ]	Any text that appears within brackets is optional and not necessary for proper sentence construction.
CAPS	If a word is written with all caps, the COP interprets the word as is. For instance, YIELD refers to an exact attribute.
lower case	A lower case spelling is a generic word. For instance, the word attribute refers to any of the data base words.
———	Underlined word(s) is necessary for a given sentence construction.
{ }	Alternatives.

as shown, states that YIELD (a specific attribute) is to be set to four when a selected attribute equals a certain input value. From the generalized example, an exact command that the COP would see is:

```
CHANGE WHERE TYPE = B52 SETTING YIELD = TO 4
```

The COP upon translating this command would set all B52s to a yield of 4.

3.2 Levels of Sentences (Syntax)

All sentences must begin with a verb. Dependent on the verb, the syntax directory defines which adverbs may be used with each verb and which verbs may have Clauses. A Clause is a level of sentence construction just below the sentence structure and is begun by an alphabetic string defined in the dictionary as an adverb. A verb may have many different clauses, and may have these clauses in any order.

The highest level of syntax within a Clause is the 'Relational Phrase,' These phrases are used to compare or set some value according to some set of rules.

Relational Phrases are constructed by 'Value Expressions' which state mathematical operation.

The most basic level of sentence syntax is the 'Value Element' such as numeric values (alphanumeric or decimal) or values of Special words.

A sentence thus consists of a verb followed by various levels of syntax. As cited, in order of decreasing levels, the syntax consists of:

- Clauses
- Relational Phrases
- Value Expressions
- Value Elements

Each level of syntax will be discussed in detail starting with the lowest level (value elements) and continuing with all levels up to the clause level. The explanation of lower sentence levels will provide an almost automatic definition of the higher levels.

3.2.1 Value Elements. The most basic item within the syntax is the 'value element' which has formats as shown in table 5. Formats under the headings of variable names, long strings and display names, are used primarily for report generation. These entries allow prints of headers, subheaders, and general display print information.

Table 5. Value Elements

FORMAT	DESCRIPTION
Alphabetic Value	Any alpha string up to six characters in length
Numeric Value	Any number, integer or decimal, signed or unsigned
Variable Names	An alphabetic string up to twelve characters in length used in a DEFINE Clause
Attributes	Any attribute name which is defined in the dictionary
Long Strings	Any alpha string of up to 120 characters in length
Special Words	Any alpha string of up to 12 characters in length defined by the dictionary as a special word
Display Names	Any alpha string up to 12 characters in length defined in the dictionary as a display name

Value definition can generally be specified by equating items to attributes. This may be as simple as directly setting outputs to attributes YIELD, VALUE, or DESIG for example. Elements may also be defined by indirect attribute reference. In other words, values may be set by requesting an attribute from a certain stated record type. If some output or calculation requires the attribute YIELD for weapon system B-52G, for example, within the allowed sentence construction, that value can be set by writing:

YIELD OF TYPE 'B-52G'

The value or end result of this request is the YIELD as input for the B-52G. The "OF" word relates the requested attribute to the record source.

3.2.2 Value Expressions. Given formats for entering values into items, the next level of syntax construction is the value expression. This is a series of value elements connected by mathematical operators and associated by parentheses.

The expression will be evaluated within a hierarchy of operations and is:

- Imbedded parenthesis; that is, all mathematical calculations will be conducted first for all operators within the highest imbedded parenthesis grouping, a value obtained, saved, and calculation continued for the next level of parenthesis and so on.
- POWER
- DIVIDED and TIMES
- PLUS and MINUS

Consider the following construction:

(YIELD TIMES (CEP PLUS 10) + 1).

Sequential steps involved are: (a) obtain the value of attribute CEP and add 10 to it; (b) to that result multiply the value of attribute YIELD; and (c) finally add one to obtain the final result. Note that value elements are set by both attributes and numeric values.

3.2.3 Relational Phrases. Value elements and/or value expressions may be connected through relational phrases. The term relational is used to refer to these phrases even when their meaning (in context) does not imply comparison. The general syntax of the relational phrase is:

subject operator object.

The subject or object may be any value element or a value expression. The subject and object may also be an ordered grouping of value elements or expressions known as a "collection" which are discussed in the next section. Allowable operators are discussed individually in following subsections.

3.2.3.1 Collections. A collection is a way of grouping subjects or objects for the purpose of easing input tasks. The collection is an ordered N-tuple, set off (or grouped as defined for value expressions) by parentheses with its elements separated by commas.

If the subject of a relational phrase is a collection, the collection is made up of either attributes or variables. Further, if the subject is a collection, the object must be a collection (or set of collections) with the value expressions being in the same order as the elements of the subject collection.

3.2.3.2 The EQUAL Operator. This operator is used either to compare the value(s) of the subject to the value(s) of the object(s) or to move the value(s) of the object(s) into the subject's value. The general form of the EQUAL phrase is:

Subject EQUAL Object [AND Object AND...AND Object]

The subject of the phrase may be a collection in which case each member of the N-tuple of the subject is compared with or set to the value of the corresponding member of the object. In the case of comparison, the logical result is true only if every comparison is true.

As can be seen, the object may be extended by the boolean operator AND (the operator OR has the precise same effect). The extension of the object is to permit the multiple comparison of the subject to several different objects or the setting of the subject to several different object values (this would create multiple records or variables). The logical implication of such a comparison is that of an OR operator - that is, if any of the comparisons (or set of comparisons) are true, the comparison is true.

A logical equivalent of an extended EQUAL with collection is shown in the example:

(YIELD, CEP) EQUAL (.7, .2) OR (.5, .1)

which is equivalent to

(YIELD EQUAL .7 AND CEP EQUAL .2) OR
(YIELD EQUAL .5 AND CEP EQUAL .1)

3.2.3.3 The GREATER and LESS Operators. These operators are strictly for comparison. The value(s) of the subject is compared to the value(s) of the object and a logical value is set as a result. If the subject is a collection the object must also be so and the logical result is true only if all comparisons are true. The comparison is strictly greater or less than.

3.2.3.4 The BETWEEN Operator. This operation is strictly for comparison. The value of the subject is compared with the closed interval whose limits are the two objects. Its format is:

subject BETWEEN object AND object

The subject values may be equal to either limit.

3.2.3.5 The LIKE Operator. This operator is used either for comparison or setting purposes. Unlike other operators, its object has a fixed format as follows:

subject LIKE attribute value

When used in this context, the attribute must be one that can specify or identify records; this format does not recognize all attributes. For instance, attributes CLASS, SIDE, TYPE, etc., are proper since they identify groupings of records, whereas attributes CEP, LONG, and GNWPNS do not. The value entry, of course, gives the value of the attribute. The subject of a LIKE phrase may be a collection but the object is not altered.

3.2.4. Clauses. Clauses follow the verb and are begun by an alphabetic string defined in the dictionary as an adverb. The syntax of every clause is defined by two qualities:

a. Clause Types

o Boolean - Sublevels of syntax are connected by boolean operators and associated parentheses. The clause will be evaluated as a logical expression from left to right or as directed via parenthetical groupings.

o Sequential - Sublevels are simply in order of entry with no implied relationship

o Single - Only one sublevel is allowed

o Null - The adverb stands alone and needs no further interpretation or modification

b. Phrase Types

- o Relational Phrases - Sublevels are used to define comparative or logical meaning according to operators
- o Restricted Relational Phrases - Used to set attributes and/or variables
- o Elements - Used to set single elements such as value elements, special words or display names.

The clause types define the overall syntax of the clause, and the phrase types define which phrases fall into or are connected by the overall syntax.

3.3 QUICKs Dictionary

All permissible words within the developed language that have meaning to the COP are contained within the 'Dictionary' (a list of tables stored in the data base). These words contain attributes (as employed for target and weapon definition), plus other words necessary for the syntax. Words within the dictionary are grouped as outlined in previous subsections. Based on these groupings, tables 6 through 11 present the entire list of words defined within QUICKs command language plus comments on how each word may be used. The list of attributes are defined in appendix A of this manual.

A sentence command written with entries not contained within the cited tables are words foreign to the language and may cause computer execution aborts. In addition, only certain combinations of words from the "dictionary" (such as verbs, adverbs, etc.) have structural meaning. The entire QUICK system generates its final output through a series of selections of individual program modules as defined by the verb. The selected module then can act (or interpret) only on those sentence patterns that request outputs produced within the program. This is also analogous to formal English where individual thoughts are expressed in separate paragraph constructions.

Many of the Adverbs in table 7 introduce clauses which are used by a number of verbs. These clauses are described in the following paragraphs.

3.3.1 DEFINE Clause. A DEFINE clause consists of a single equals phrase in which the subject is used as an alphabetic variable name. (The subject should not be an attribute.) The object of the phrase should be a mathematical formula combining attributes and numeric values plus any alphabetic string which the user intends to employ as the variable name of this or any other DEFINE.

Table 6. QUICKs Text English Verbs

VERB	MODULE	DESCRIPTION
ALTER	REPORT	Makes updates and changes to a previously designed display
ASSIGN	JLM	Builds the Assignment table
ASTERISK	JLM	Makes Damage Assessment Tape from a JAD format tape and the Data Base
BUILD	EIM	Instructs the External Interface Module to build a file
CHANGE	DATA	Updates data element(s)
CREATE	DATA	Creates new data elements
DELETE	DATA	Deletes records
DESIGN	REPORT	Constructs and saves a REPORT Module display
EDIT	EDITDB	Executes the EDITDB module
INDEX	INDEXER	Executes the INDEXER module
MODIFY	DBMOD	Executes the DBMOD module
PLANSET	PLANSET	Executes the PLANSET module
PLOTDATA	EIM	Creates plot tapes
PREPARE	PREPALOC	Executes the PREPALOC module
PRINT	REPORT	Prints some user defined (built by DESIGN) display
RESTORE	SRM	Brings the contents of an IDS data base from magnetic tape to a disk file
SAVE	SRM	Stores the contents of an IDS data base on a magnetic tape
SELECT	JLM	Selects records from a JAD file

Table 7. QUICKs Adverb List[*]
(Part 1 of 2)

ADVERB	DESCRIPTION
ALPHAS	Defines the CLASS, TYPE, DESIG, and category relationships for the Assignment table
ATTACKERS	Defines the attacking weapon systems
DEFENDERS	Defines the exemplar targets and associated values
DEFINE	Describes a user defined variable for use in either printed or tape/file output
DISPLAY	Names a REPORT module display
FIELDS	Lists attributes in the data base to be tested to see if they are within defined ranges
FILE	Used to define type of File EIM should build
FIX	Defines weapon fix assignments
FORMAT	Describes the format to be used in the creation of printed or tape/file output
KEEPING	Lists the DESIGs which are not to be deleted from the data base after ASTERISK is run
ONPRINTS	A request to print non-standard reports
OMITTING	Keeps duplicate targets from being added to the data base.
ORDER	Allows the user to input the order that the classes will be entered into the data base
PLAYERS	Describes valid country codes and the regions that they are in
PRIORITY	Introduces a complexing priority scheme
REPLACING	Causes duplicate targets to be replaced in the data base
SAME	Describes a record similar to the one being created which is to be used for default values
SETTING	Introduces a clause that stores data elements
SORT	Describes the sort order for printed or tape/file output
SUPRESSING	Suppresses data value editing during data creation

[*] See table 8 for proper usage.

Table 7. (Part 2 of 2)

ADVERB	DESCRIPTION
UNIT	Gives the tape/file logical unit number
USING	Requests data value editing during data creation
WHERE	Describes the subset of the data base on which the verb's action is to be performed
WITH	Describes the relationships that must be met between selected attributes in the data base
VNOPTION	Used to select a complexing option

Table 8. QUICKs Text English Adverbs Usage

ADVERB	CLAUSE TYPE	PHRASE TYPE	VERB(S)
ALPHAS	Sequence	Elements	ASSIGN
ATTACKERS	Sequence	Elements	PLANSET
DEFENDERS	Sequence	Restricted Relational	PLANSET
DEFINE	Single	Restricted Relational	DESIGN, ALTER, BUILD
DISPLAY	Sequence	Elements	DESIGN, PRINT, ALTER
FIELDS	Sequence	Elements	EDIT
FILE	Single	Element	BUILD
FIX	Sequence	Restricted Relational	PREPARE
FORMAT	Sequence	Elements	DESIGN, ALTER, BUILD
KEEPING	Sequence	Elements	ASTERISK
ONPRINTS	Null	Null	(ALL VERBS)
OMITTING	Null	Null	SELECT
ORDER	Sequence	Elements	SELECT
PLAYERS	Sequence	Elements	ASSIGN
PRIORITY	Sequence	Elements	PLANSET
REPLACING	Null	Null	SELECT
SAME	Sequence	Elements	CREATE
SETTING	Sequence	Restricted Relational	PLANSET, MODIFY, DESIGN, PREPARE, CREATE, ALTER, CHANGE, PLOTDATA, SELECT
SORT	Sequence	Elements	DESIGN, ALTER, BUILD
SUPRESSING	Null	Null	CREATE
UNIT	Single	Element	SELECT, SAVE, RESTORE, BUILD
USING	Null	Null	CREATE
WHERE	Boolean	Relational	DESIGN, PRINT, ALTER, CHANGE, DELETE, SELECT, BUILD, EDIT
WITH	Sequence	Relational	INDEX, EDIT
VNOPTION	Null	Null	INDEX

Table 9. QUICKs Text English Special Words
(Part 1 of 2)

SPECIAL WORD	USE	DESCRIPTION
A	SORT adverb	Same as ASCENDING
AFTER	ALTER verb, FORMAT adverb	Introduces additions to format after indicated PAGE, LINE, etc.
ASCENDING	SORT adverb	Lowest values will be first
D	SORT adverb	Same as DESCENDING
DESCENDING	SORT adverb	Highest values will be first
HEADER	FORMAT adverb	Introduces title which appears at top of each page
IN	SELECT verb	Restricts the assignment of a target based on the country it is located in.
	FORMAT adverb	Introduces user specified data item output format (e.g., I4)
LINE	FORMAT adverb	Introduces format for body of report
NEW	DESIGN verb, DISPLAY adverb	Indicates display name is new
OLD	DESIGN verb, DISPLAY adverb	Indicates new display is to replace previous display with same name
OTHER	BUILD verb, FILE adverb	Introduces user defined output tape/file
OWNED	ALPHAS adverb	Restricts assignment of a target based on the country code of its owner
PAGE	FORMAT adverb	Introduces new set of HEADERs, TRAILERs and LINEs
PAGENO	FORMAT adverb	Indicates page number is to be displayed
REMOVE	ALTER verb, FORMAT adverb	Indicates named PAGE, LINE, etc. is to be removed
REPLACE	ALTER verb, FORMAT adverb	Introduces format items to replace indicated PAGE, LINE, etc.
SIDAC	BUILD verb, FILE adverb	Indicates output tape/file is to be produced for SIDAC
SPACES	FORMAT adverb	Indicates a series of blank spaces in the format

Table 9. (Part 2 of 2)

SPECIAL WORD	USE	DESCRIPTION
TABLE	FILE adverb	Indicates output tape/file is to be produced for TABLE
TRAILER	FORMAT adverb	Introduces format which is to appear at the bottom of each page
X	FORMAT adverb	Same as SPACES

Table 10. QUICKs Text English Operators

ALPHABETIC	SPECIAL CHARACTER	DESCRIPTION
	,	Comma - used in collections and other formats
	(	Left Parenthesis - used to introduce collections and for associative purposes in boolean and mathematical expressions
	)	Right Parenthesis - used to end collections and for associative purposes in boolean and mathematical expressions
AND	&	Boolean operator - also used to extend EQUALS relational expressions
OR		Boolean operator - also used to extend EQUALS relational expressions
NOT		Boolean operator
EQUAL or EQUALS	=	Relational operator (two spellings permitted: EQUAL or EQUALS)
GREATER THAN	>	Relational operator - strictly greater than
LESS THAN	<	Relational operator - strictly less than
BETWEEN		Relational operator - defines a closed interval
LIKE		Relational operator
PLUS	+	Mathematical operator (symbol + may also be used as a sign)
MINUS	-	Mathematical operator (symbol - may also be used as a sign)
TIMES	*	Mathematical operator
DIVIDED BY	/	Mathematical operator
TO THE POWER	**	Mathematical operator
OF		Special operator used to modify attributes

Table 11. QUICKs Text English Nulls

NULL	DESCRIPTION
AS	--
BY	--
DIRECTORY	Used to check data element limits
DUPLICATES	Associated with SELECT verb (JLM) to ignore or replace duplicate records
FOR	--
FROM	--
LOCATED	Used with ASSIGN verb; i.e., LOCATED IN (a special word)
ON	--
THAN	--
THE	--
TO	--

There are four types of DEFINES:

- o Normal - Contains only other normal defines
- o Sum - Contains its own name as part of an arithmetic sum
- o Product - Contains its own name as part of a formula which is a product or power
- o Special - Contains another define which is not a normal define

3.4.1.1 Normal DEFINE. This is a DEFINE that is determined anew for each requested record. To explain, consider the command:

```
DEFINE PROB = REL * 100
```

For every record where PROB is requested, the calculation given will be determined. This differs from other types to follow.

3.4.1.2 Sum DEFINE. This DEFINE is a cumulative result of all records processed. Consider:

```
DEFINE TOTALVAL = TOTALVAL + VAL*10
```

Note that the subject appears also on the right hand side of the equals operator. This and the appearance of the plus operator makes this a type sum DEFINE. Simply it means to accumulate into the defined variable name (TOTALVAL) the result of the last calculated value for the variable name and add the result of the included mathematical expression for the record currently being processed. In the mathematical sense, TOTALVAL can be written as:

$$\text{TOTALVAL} = \Sigma \text{VAL}*10$$

where the summation is over all records processed.

3.4.1.3 Product DEFINE. This is the same as the Sum DEFINE except that the Product DEFINE is cumulatively multiplied for each processed record. Consider:

```
TOTALVAL=TOTALVAL*VAL*10
```

3.4.1.4 Special DEFINE. This type DEFINE is a combination of either the sum or product DEFINE plus a second definition. It is calculated only once but its internal variables can only be determined after other DEFINES are calculated. Using the example of the variable name TOTALVAL as cited for the sum type DEFINE, a special DEFINE could be:

```
DEFINE NEWVAL = TOTALVAL/100
```

This DEFINE is calculated only once but it included a DEFINE that is calculated for each record processed.

3.3.2 FORMAT Clause. The FORMAT clause is used by both the REPORT and EIM modules to define the format of output tapes, files and printed reports. The FORMAT clause consists of a series of subclauses introduced by one of the four special words: PAGE, LINE, HEADER, and TRAILER.

3.3.2.1 PAGE. This special word begins a set of headers, trailers and lines. Each set is produced separately. The PAGE special word has no elements following it. A FORMAT clause need not be started by a PAGE special word; one will be assumed if the user omits it.

3.3.2.2 HEADER. This special word introduces a string of elements that define one or more lines which are to appear at the top of every physical page. The headers will be produced in the order input.

3.3.2.3 TRAILER. This special word introduces a string of elements that define one or more lines which are to appear at the bottom of every physical page. Trailers will be produced in the order input.

3.3.2.4 LINE. This special word introduces a string of elements that make up the body of the report or file. Lines are of two types: those that contain attributes and DEFINEs that are not sums or products and those which contain no attributes and only DEFINEs which are sums or products. The former will produce one line for every logical record retrieved from the data base; the latter will produce only one line.

3.3.2.5 Special Words and Elements. HEADER, TRAILER and LINE special words are followed by a series of elements or element sets which may be any of the following:

- o Alphabetic - used as a constant as input

- o Attribute - an attribute will be displayed either in a format calculated by the module or provided by the user via an IN phrase

- o Define variable name - a variable will be displayed either in a format calculated by the module or provided by the user via an IN phrase

- o IN special word - this special word must follow an attribute or variable name and must be followed by a FORTRAN format (viz. F10.6). In this way the user specifies the format in which the attribute or variable is to be displayed

- o Long String - an alphabetic string enclosed by either of the characters (' or ") appears in the format as input

- o Numeric - followed by the special word SPACES (or X) produces as many blank spaces as is specified by the numeric.

3.3.3 ONPRINTS Clause. This is the general clause used to select the optional print of any verb.

3.3.4 SETTING Clause. This clause is used to specify values for attributes. It may contain either equals or like phrases. Many modules will restrict the attributes which may be used. The user should be aware that the use of an extended equals (e.g., (LAT,LONG)=(50,130) & (60,135) & (70,120) has special meaning within the context of the CHANGE verb.

3.3.5 SORT Clause. This clause is used to specify the sort order to be used in the case of an output file or report. The clause consists of a series of pairs of elements. The first element of the pair is either an attribute or a defined variable name. The second element of the pair is one of the two special words: ASCENDING (or A) or DESCENDING (or D). The order of the sort is the order of the element pairs, the first pair specifying the major sort. An example of a sort clause is:

SORT LAT ASCENDING LONG DESCENDING

3.3.6 UNIT Clause. This clause is used to specify a logical tape or file unit number. It has the form:

UNIT logical unit number

3.3.7 WHERE Clause. This clause is used to specify a subset of the data base for processing by the module called. It has the form:

WHERE a_1 [AND a_2 AND a_3 AND . . . AND a_n]

or

WHERE a_1 [OR a_2 OR a_3 OR . . . OR a_n]

where the a_i are logical expressions. The simplest logical expression is a relational phrase. However, a logical expression can also be of the form: NOT a_i

where a_i is a logical expression. This form implies the negation of the expression a_i. Finally, a logical expression may be of the form:

WHERE (a_1 [AND a_2 AND . . . AND a_n])

or

WHERE (a_1 [OR a_2 OR . . . OR a_n])

where the a_i are logical expressions

The user should be aware that the use of collections and/or extended EQUALS (i.e., (LAT, LONG) = (50,150) & (60,160)) creates internally a complex logical expression and, if imbedded within another complex logical expression, should be enclosed in parentheses.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 4. CENTRAL OPERATIONS PROCESSOR

4.1 General Purpose

The Central Operations Processor (COP), or QUICKs executive software module, acts as an intermediary between the user and the operating system, and provides the communication between the QUICK system and the integrated data base. The COP acts accordingly by interpreting user commands (explained in section 2 of this manual) written in imperative text English formats.

Inputs to the COP are completely generalized to the degree that the text English commands are. The COP itself generates very little output which is one of the main functions of each module that the COP executes. Standard inputs and outputs are discussed within the following subsections.

4.1.1 BOOT Entry. Given an integrated data base, the COP operates free form. However, the data base must be initially created as to the structure and specifications of data items within the structure. Since the COP must operate with the integrated data base, a special entry (or mode of execution) is necessary to inform the COP that data follows whereby the data base itself will be defined. This mode of execution is performed within the BOOT module which is a special portion of the COP. The BOOT module will read data and create the data base structure. This creation is a one-time operation and once successfully executed, only minor updates will be necessary. Separate subsections, following the discussion on the COP, will outline the BOOT module operation.

4.2 Inputs to the COP

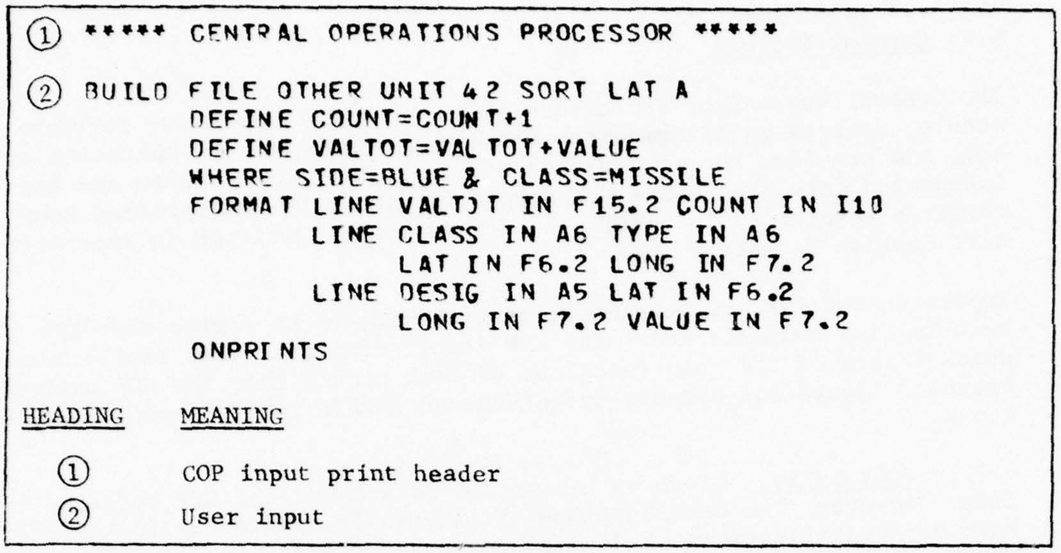
There is no standard input to the COP, only the previously cited generalized text English commands. From the verb within each command, the COP directs execution of each related module. Each module within this, and other user manuals will further detail the nature of the inputs.

Appendix E of this manual presents the Job Control Language (JCL) necessary to execute the COP for batch submittals as well as a series of commands for timesharing submittals.

4.3 COPs Output

The standard output of the COP is a print of the input commands preceeded by an introductory header (figure 3). Further, the standard and/or optional output of each module is preceeded by a banner print of the module's overlay link name (figure 4). This banner appears each time a new module is called and will only appear before the first instance of consecutive calls to the same module.

The COP error messages appear in figure 5.



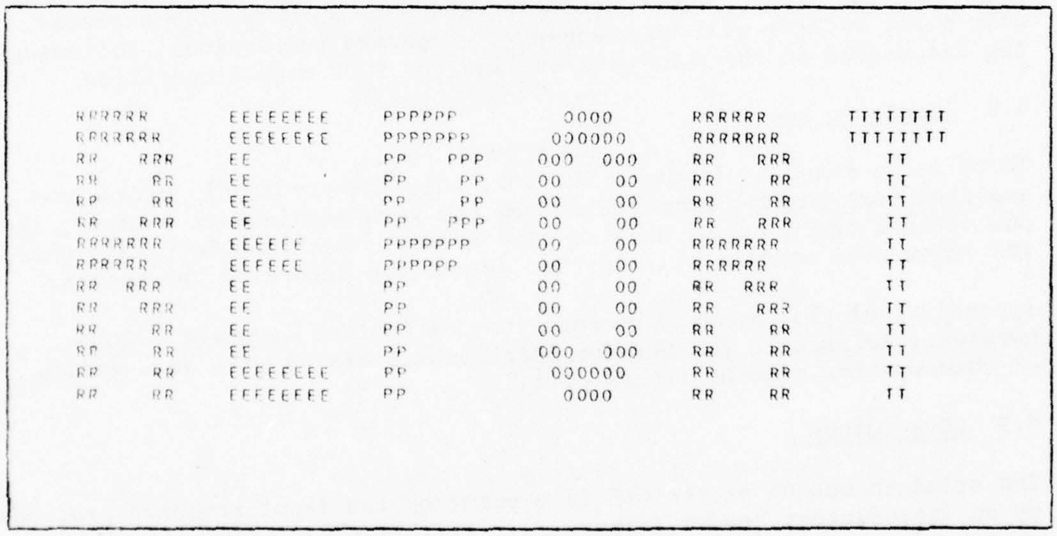
```
① ***** CENTRAL OPERATIONS PROCESSOR *****

② BUILD FILE OTHER UNIT 42 SORT LAT A
        DEFINE COUNT=COUNT+1
        DEFINE VALTOT=VALTOT+VALUE
        WHERE SIDE=BLUE & CLASS=MISSILE
        FORMAT LINE VALTOT IN F15.2 COUNT IN I10
               LINE CLASS IN A6 TYPE IN A6
                    LAT IN F6.2 LONG IN F7.2
               LINE DESIG IN A5 LAT IN F6.2
                    LONG IN F7.2 VALUE IN F7.2
        ONPRINTS
```

HEADING MEANING

① COP input print header

② User input

Figure 3. Central Operations Processor, Input Print



```
RRRRRR        EEEEEEEE     PPPPPP          0000        RRRRRR         TTTTTTTT
RRRRRRR       EEEEEEEE     PPPPPPP        000000       RRRRRRR        TTTTTTTT
RR   RRR      EE           PP   PPP      000  000      RR   RRR          TT
RR    RR      EE           PP    PP      00    00      RR    RR          TT
RR    RR      EE           PP    PP      00    00      RR    RR          TT
RR   RRR      EE           PP   PPP      00    00      RR   RRR          TT
RRRRRRR       EEEEEE       PPPPPPP       00    00      RRRRRRR           TT
RRRRRR        EEEEEE       PPPPPP        00    00      RRRRRR            TT
RR  RRR       EE           PP            00    00      RR  RRR           TT
RR   RRR      EE           PP            00    00      RR   RRR          TT
RR    RR      EE           PP            00    00      RR    RR          TT
RR    RR      EE           PP            000  000      RR    RR          TT
RR    RR      EEEEEEEE     PP             000000       RR    RR          TT
RR    RR      EEEEEEEE     PP              0000        RR    RR          TT
```

Figure 4. COP Banner Print

1 INPUT EXCEEDS MAXIMUM ALLOWED

For a given verb, there are too many sublevels within the sentence. Restructure the sentence into several commands.

2 $*$*$ QDATA ENCOUNTERED (A6) ERROR WITH ARGUMENT (A6) $*$*$

IDS error. First argument is either standard IDS error code or one of the three special QUICK codes. (Refer to IDS User Manual.)

OOORRR - Illegal record type
OOOCCC - Illegal chain name
OOOILC - Illegal call

Second argument is name of record type or chain which caused error. Note that a code of 000S01 means the IDS file has run out of space.

3 SYNTAX ERROR - COMMAND ENDED TOO SOON

End of input sentence occurred where illegal

4 SYNTAX ERROR - VERB NOT RECOGNIZED

Error in syntax directory, verb recognized by dictionary not in syntax directory

5 SYNTAX ERROR - ADVERB OUTSIDE COMMAND

Adverb found where verb expected.

6 SYNTAX ERROR - CLAUSE MET EARLY END

End of clause where not expected.

7 SYNTAX ERROR - LOST ITEM

Item, which is neither verb nor adverb, where one was expected.

8 SYNTAX ERROR - ATTRIBUTE IN VALUE ELEMENT NOT ID

Error in OF phrase, identifier attribute not defined as such in dictionary.

9 SYNTAX ERROR - VALUE ELEMENT WRONG

Error in syntax of value element.

Figure 5. COPs Error Messages (Part 1 of 3)

10 SYNTAX ERROR - UNBALANCED PARENS IN VALUE EXPRESSION

Error in mathematical statement-unbalanced parentheses.

11 SYNTAX ERROR - RELATIONAL EXPRESSION HAS BAD COLLECT

Unbalanced collection or collection containing illegal items.

12 SYNTAX ERROR - ILLEGAL RELATIONAL OPERATOR

GREATER, LESS or BETWEEN appears in clause restricted to EQUAL and LIKE.

13 SYNTAX ERROR - ERROR IN SUBJECT OF RELATIONAL PHRASE

Subject should be either an attribute or define name.

14 SYNTAX ERROR - NO ID-ATTRIBUTE FOR LIKE

LIKE phrase has no identifier attribute.

15 SYNTAX ERROR - LIKE PHRASE HAS NO VALUE

Value for Identifier attribute is missing.

16 SYNTAX ERROR - ABNORMAL END TO BOOLIAN

Boolean clause ended where illegal.

17 SYNTAX ERROR - BOOLIAN STATEMENT IN WRONG ORDER

Elements of boolean clause follow one another in an illogical fashion.

18 SYNTAX ERROR - ILLEGAL ITEM IN VALUE EXPRESSION

Value expression contains illegal item.

19 SYNTAX ERROR - ILLEGAL ELEMENT

Item included in clause of elemental adverb which is illegal according to syntax directory.

20 SYNTAX ERROR - TOO MANY PHRASES

Adverb defined as "single" has more than one phrase.

Figure 5. (Part 2 of 3)

21 SYNTAX ERROR - RELATION NOT RIGHT IN SEQUENCE

Normal sequence of relational phrase violated.

22 TYPE (I5) VALUE (I5) (A12)

Arguments are the type, value and alphabetic representation of the offending item. This message appears following most of the messages that begin with "SYNTAX ERROR"

23 INPUT TABLES EXCEEDED, TYPE (I2)

Tables built by subroutine ERRFND have been exceeded. Argument indicates type of table in which error occurred.

24 LONG STRING TOO LONG

Input long string exceeds 120 characters.

25 UNBALANCED COLLECTION

Number of items in object collection(s) does not agree with number in subject.

26 BOOLEAN STATEMENT WRONG

Error in boolean statement logic.

Figure 5. (Part 3 of 3)

4.4 BOOT Module

4.4.1 General Purpose. The BOOT Module is designed to create and update those portions of the data base which are essential to normal COP operation. As a result, the input to BOOT is on fixed formatted card images rather than a free form text English input. The portions of the data base which BOOT addresses are:

- The data organization index, which contains a functional description of the IDS data base strucutre, the information required to retrieve headers and the data editing directory
- The data entry point headers
- The dictionary
- The text English syntax directory
- The module link table
- The target longitude sectors

In general, each input card image to BOOT creates or updates a record within one of the above structures. Normally, BOOT will be run only when a QUICK data base is being created from scratch.

4.4.2 Input. The input to BOOT consists of an introductory command verb followed by a series of sets of card images. Each set is begun by a command adverb and terminated by a card containing END in card columns 1-3. The last set is followed by a second END card. The order of the sets is important if the user is building a data base from scratch because the creation of some records is dependent upon the previous creation of others. The order to be followed is that in which the sets will be presented.

4.4.2.1 Verb Command. The introductory verb for BOOT is INITIALIZE. This command must be defined starting with column 1 on the first card image that the COP reads in.

4.4.2.2 Introductory Adverbs. As cited, each set is introduced by a command adverb and ended with an END card. The command adverbs must appear on separate card images with the adverb starting in column 1. The command adverbs and the general description of the sets they introduce are shown in table 12. Discussions of each set follow.

4.4.2.3 NEWINDEX Adverb. This command adverb has no following cards. It must appear only in the case of a construction of the data base from scratch. In this case it must appear first. It causes the creation of the data organization index header and a utility table header used internally by COP.

Table 12. BOOT Command Adverbs

NEWINDEX	Creates new index header
RECORDTYP	Adds new record type records
HEADER	Adds data entry point headers
DICTIONARY	Makes entries and updates to the dictionary
MODULE	Makes changes to the module link table
SYNTAX	Makes entries and changes to the text English syntax directory
INDEX	Makes entries and changes to the data organizational index
SECTORS	Adds new sector records

4.4.2.4 RECORDTYP Adverb. A complete and total definition of each record type contained within the integrated data base is presented in Program Maintenance Manual Volume I. This command adverb introduces the list of those record types to be added to the data organizational index. Each card image following the adverb contains the name and number of the record type. For each card image, the record type name is placed in columns 1 to 8 (left justified) and the record type number in columns 9 to 16 (free field). The set is ended normally with the insert of the END card.

4.4.2.5 HEADER Adverb. In the IDS sense, a header is the highest level of entry into a series of master records and their associated chains. In the QUICK data base, headers define entries to data elements through the use of attribute CLASS. This set of commands, then, specifies header name (or class entries) and, further, links them to record type as outlined above.

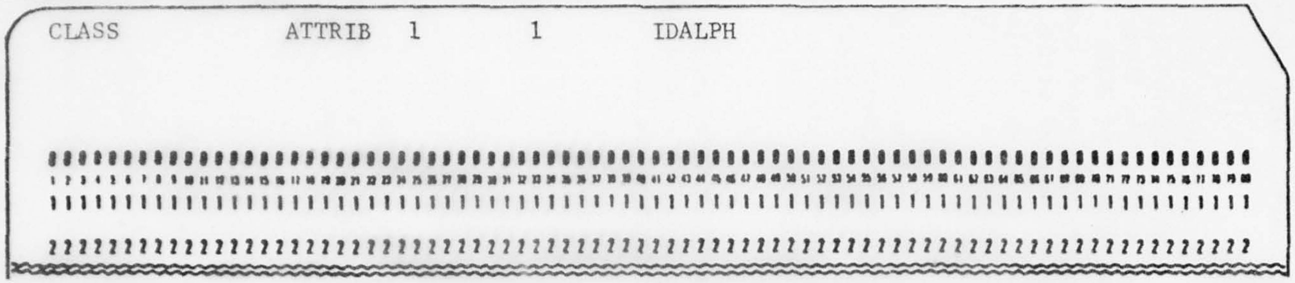
Each card in the set defines the record type name (columns 1 to 8, left justified), the attribute CLASS name (columns 9 to 16), the attribute SIDE name (columns 17 to 24). SIDE may be left blank. Also, if the header record type is TGTHD (target header), the card image may contain an entry for ICLASS (columns 25 to 32) which is used as an internal sequential counter. This set is ended by an END card.

4.4.2.6 DICTIONARY Adverb. All of the words that have meaning to the COP are contained within the dictionary (see appendixes A, B, and C) and are added or altered with this command adverb. The card image format is shown in figure 6. The word types defined with the dictionary are:

- OPERAT - text English mathematical, relational, boolean or syntax operator
- VERB - text English verb
- ADVERB - text English adverb
- SPCIAL - text English special word
- ATTRIB - attribute
- NULL - text English null

If the word is an attribute, the various type identifications are:

- NMALPH - alphabetic attribute
- IDALPH - alphabetic attribute used as an IDS identifier
- NMNUMR - numeric attribute
- IDNUMR - numeric attribute used as an IDS identifier



Columns	Meaning
1-16	Word (left justified). See section 3 for text English words, appendix A for attributes.
17-24	Dictionary Word Type Mnemonic (left justified). See subsection 4.4.2.6.
25-32	Word identifying number (free field). See appendixes C and D. No entry required for nulls.
33-40	Attribute common block address (free field). See appendix C.
41-48	Attribute type Identification mnemonic (left justified). See subsection 4.4.2.6.
49-80	Not used

Figure 6. Dictionary Card Image

Each card image causes the current dictionary to be queried for existence of each word. If it exists, the information concerning the word is updated; otherwise, the new word is added to the dictionary. This set is ended by an END card.

4.4.2.7 MODULE Adverb. When the COP encounters a command verb, it executes a module, or that is to say it calls into memory an overlay which will in turn carry out the verb's instruction. Each verb, then, corresponds to a module (see section 3). The MODULE adverb of BOOT provides the method of connecting the verbs and module. One card image is required for a verb-module combination. The verb is placed within columns 1 to 16 (left justified) and the link name (or module) is placed within columns 17 to 24 (left justified). This set is ended by an END card.

4.4.2.8 SYNTAX Adverb. This command adverb introduces the set of card images which create and/or alter the text english syntax directory or the grammar. This directory contains four distinct types of records each of which is created by a card image with a record type name in columns 1-6; these are: SYNVRB, PRMADV, ADVELM, and SYNCLZ. Each record type is discussed in the order of appearance.

a. SYNVRB -- This card image creates an entry for a verb and tells the syntax analyzer whether or not it should expect clauses to follow the verb (see section 3). Each card image has the form:

Columns	Meaning
1-6	=SYNVRB
9-24	Verb (left justified). See section 3.
25-32	=CLAUSE, verb must be followed by clause(s) =NOCLAU, verb need not be followed by a clause

b. PRMADV -- This card image creates an entry for an adverb and describes to the syntax analyzer the type of clause and phrases which follow the adverb. Each card image has the form:

Columns	Meaning
1-6	=PRMADV
9-24	Adverb (left justified). See section 3.
25-32	Clause type mnemonics: =BOOL, Boolean clause. Clause phrases are connected by boolean operator =SEQENC, Sequence clause. Clause phrases are not connected in any way =SINGLE, Clause has one and only one phrase =NONE, Clause has no phrases

Columns	Meaning
33-40	Phrase type mnemonics (left justified): =RELATE, Phrases are relational phrases =EQULIK, Relational operator EQUAL(=) and LIKE only may be used =ELEMNT, Phrases are language elements (i.e., number, long string, special word, etc.)

c. ADVELM -- This card image is used to describe to the syntax analyzer the types of language elements which are legal in the case where the adverb has been described as having elemental phrases. The card format is:

Columns	Meaning
1-6	=ADVELM
9-24	Adverb (left justified). See section 3.
25-32	Either element or word type mnemonic (left justified). If element: =ALPNUM, Alphanumeric value =NUMBER, Numeric value =LONGST, Long string (alphabetic value greater than 12 characters) If word type mnemonic, enter values as given in subsection 4.4.2.6.
33-40	Operator or special word <u>value</u> (left justified). See section 3.

d. SYNCLZ -- This card image is used to inform the syntax analyzer which adverbs go with which verbs. Any verb may be related to as many adverbs as desired and vice versa. Each card image has the form:

Columns	Meaning
1-6	=SYNCLZ
9-24	Verb (left justified). See section 3
25-32	Adverb (left justified). See section 3

The set is ended by an END card.

4.4.2.9 INDEX Adverb. Through multiple entries to the integrated data base, the COP has the ability to honor all reasonable user requests. In addition to the entries, each data record is connected in some logical fashion whereby the COP may conduct queries in orderly schemes. However, the COP needs to know the structure of the data base linkage. Therefore, a network must be built which in turn tells the COP how the integrated data base 'looks' in the sense of what is chained to what and which masters (an IDS type record) have masters under them. The INDEX adverb will do this. For all discussion below, refer to Program Maintenance Manual Volume I for a total picture of the data structure.

Under the command adverb INDEX there are five record types: INDMST, INDDET, INDATR, ALPHVL and LINKER. Each record type is discussed.

a. INDMST -- A card image of this type should be included for each chain in the data base. It contains the information necessary to determine what chains each record type is a master of and the detail record type of that chain. Each card image format is:

Columns	Meaning
1-7	=INDMST
9-16	Master record type name (left justified)
17-24	Detail record type name (left justified)
25-32	Chain name (left justified)

b. INDDET -- A card image of this type is needed for each chain in the data base. It contains the information necessary to determine what chains each record type is a detail of and the master record type of that chain. The card image format is:

Columns	Meaning
1-6	=INDDET
9-16	Detail Record type name (left justified)
17-24	Master record type name (left justified)
25-32	Chain name (left justified)

c. INDATR -- A card image of this type is needed for each attribute in the data base. It contains information as to the attribute type and editing criteria such as default values and upper and lower values for numeric attributes. Appendix B details current directory specifications. The card format is:

Columns	Meaning
1-6	=INDATR
9-24	Attribute Name (left justified)
25-32	Attribute type mnemonic (left justified)
33-40	Default value (alphabetic values should be left justified
41-48	Lower limit (free field)
49-56	Upper limit (free field)

d. ALPHVL -- This card image creates the list of legal values for attributes whose type mnemonic is some combination of 'LIST' values. Each card image is:

Columns	Meaning
1-6	=ALPHVL
9-24	Attribute name (left justified)
25-32	'LIST' values (left justified)

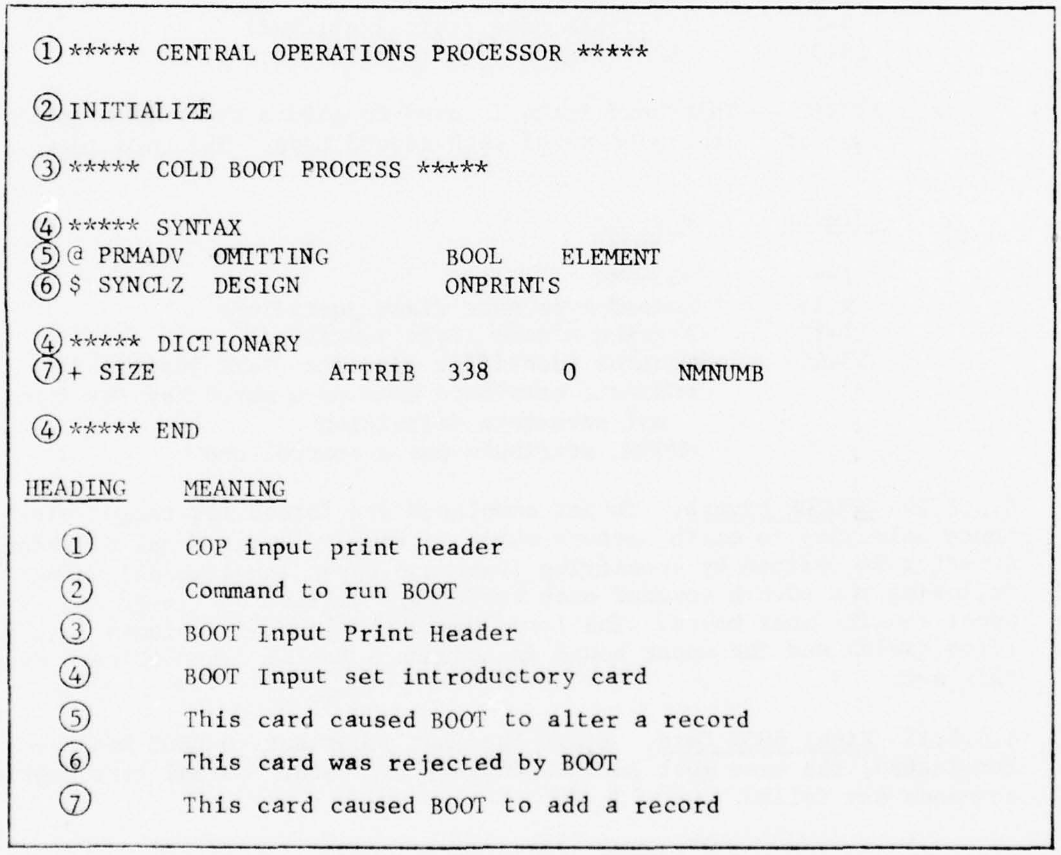
e. LINKER -- This card image is used to inform the data organization as to the make up of each record type. The card image is:

Columns	Meaning
1-6	=LINKER
9-16	Record type name (left justified)
17-32	Attribute name (left justified)
33-40	Control identifier mnemonic (left justified)
	=CNTROL, attribute used as a match key for internal structure definition
	=NORM, attribute not a control one

4.4.2.10 SECTOR Adverb. Target complexes are formed for target elements belonging to earth sectors which is simply longitudinal division. A sector is defined by specifying lower and upper longitudinal values. Following the adverb command each card image defines one lower and upper longitudinal bound. The lower bound is placed in columns 1 to 8 (free field) and the upper bound in columns 9 to 16. The END card ends this set.

4.4.2.11 Final BOOT Card. After the last input set to BOOT has been terminated, the user must include another END card. Normal text English commands may follow.

4.4.3 Output. The BOOT module produces an annotated list of its input on print report code 13. Each card image as input is displayed as input. (Columns 73-80 are also printed so as to provide any sequencing information the user desires to place there.) Command adverbs are given a special flag. In addition, each card image is preceded by a +, @ , or $ symbol. Respectively, these symbols imply a record type was added, altered or an error occurred. A sample of portion of the BOOT module output appears in figure 7.



```
①***** CENTRAL OPERATIONS PROCESSOR *****

②INITIALIZE

③***** COLD BOOT PROCESS *****

④***** SYNTAX
⑤@ PRMADV  OMITTING         BOOL    ELEMENT
⑥$ SYNCLZ  DESIGN           ONPRINTS

④***** DICTIONARY
⑦+ SIZE             ATTRIB  338     0        NMNUMB

④***** END
```

HEADING	MEANING
①	COP input print header
②	Command to run BOOT
③	BOOT Input Print Header
④	BOOT Input set introductory card
⑤	This card caused BOOT to alter a record
⑥	This card was rejected by BOOT
⑦	This card caused BOOT to add a record

Figure 7. BOOT Module Output Print

SECTION 5. DATA MODULE

5.1 General Purpose

The DATA module is designed to allow the user to create those portions of the data base not created by other modules, add information to existing record types which were only partially initialized by other modules, make corrections and updates to existing record types and delete record types which are no longer desired. The DATA module has three verbs: CREATE, CHANGE, and DELETE. The general philosophy of the DATA module is to allow the user to make commands solely in terms of attributes, relieving him of the necessity of knowing which record type or types are affected. Also, the execution of the DATA module is permissible at any stage of processing within the QUICK system.

5.2 Input

5.2.1 The CREATE Verb. The CREATE verb defines new data record types. The DATA module builds record types to be created filling in any unsupplied attribute values either from QUICKs directory defaults or from a user specified source. In general, the use of the CREATE verb will cause one or more sets of record types to be created. DATA also will check all input values against QUICKs directory for validity. Finally, the DATA module checks to make sure that no duplicate records are created such as two separate target records having the same DESIG.

If the user desires to create targets via DATA he does not necessarily have to specify values for attributes TYPE, DESIG, TASK, or any other attribute value normally obtained if the JLM has successfully built an Assignment table (see Users Manual Volume II, section 2). These values will be supplied from the Assignment table where not user specified.

The CREATE verb has five legal adverbs:

o SETTING - defines values to be stored

o SAME - alternate method of storing default attribute values

o USING - specifies the use of QUICKs directory

o SUPRESSING - specifies that QUICKs directory is not to be used

o ONPRINTS - optional request for printing data of interest to a maintenance programmer

A sample CREATE verb input could be:

```
CREATE SETTING CLASS=BOMBER SIDE=BLUE
(IREG, LAT, LONG) = (1,65,150) & (2,63,51) &
(3,63,3) ONPRINTS USING DIRECTORY
```

This command creates three records for the given CLASS and SIDE where each record has the input values for IREG, LAT and LONG. Note that TYPE, TASK, and CNTRYLOC are not given; they are assumed to be obtained from a created Assignment table.

Comments on the major use of adverbs follows.

5.2.1.1 The SETTING Adverb. This is the chief clause for the CREATE verb. It is that part of the command structure whereby the user informs the COP what values are to be stored and under what record type. Any attribute may be set through this adverb. One setting clause will create one record type except if the clause contains one or more extended equals phrases (i.e., LAT=40 & 50 & 60) in which case a record type set will be created for every combination of the values in such phrases. Note that a create verb may contain any number of SETTING clauses.

5.2.1.2 The SAME Adverb. This clause allows the user to specify a logical record which is used to define the default values for attributes which the user has not input. Its purpose is to reduce the inputting of redundant information. For instance, say the user desires to create two weapon systems, one for a B52 and another for a B52A and both systems have identical attribute values except that the attribute RANGE differs between the two. Under this scenerio, the user would have to input two complete sets of identical attribute values if not for the SAME clause. Assume the B52 record has been created and all of its attributes stored. A command, then, to create the B52A record could be:

```
CREATE SETTING TYPE=B52A RANGE=9000
       SAME AS TYPE B52
```

Now the B52A record type is created identically to the B52 but with a different value for RANGE.

It is appropriate to show examples of the LIKE and OF operation which are helpful in creating data. Consider the command:

```
CREATE SETTING TYPE=B52A RANGE LIKE TYPE F111
       SAME AS TYPE B52
```

Type B52A will have all of the same attribute values as the B52 except that the RANGE for the B52A will equal the range of the F111.

An example using the OF operator with the CREATE command would be:

```
CREATE SETTING CEP=10*CEP OF TYPE B58 . . .
```

The CEP to be created equals 10 times the CEP value stored for TYPE B58.

The general form for the SAME adverb is:

```
SAME AS Identifier attribute Identifying value
```

The combination of attribute and value is used to retrieve a record of the type defined in the SETTING clause where the identifier attribute has the identifying value. The attribute defaults are then retrieved. The attribute must be an attribute identified in the dictionary as being a unique identifying attribute (e.g., CLASS, TYPE, etc.). If this clause is omitted, the default values used for attributes not provided values in the SETTING clause will be those defaults found in QUICKs directory.

5.2.1.3 The SUPPRESSING and USING Adverbs. These adverbs are used to specify whether or not the DATA module is to use QUICKs directory to edit input values. If not specified, USING is assumed and all values are edit checked.

5.2.2 The CHANGE Verb. The CHANGE verb is used when the user wishes to alter existing record types. Information following the verb must be sufficient to inform the COP of what record type(s) is to be affected and what attribute values are to be changed. A WHERE clause is used for specification of the record types to be changed and a SETTING clause gives the altered attribute values. In addition to the standard optional ONPRINTS adverb, these are the only two adverbs recognized by the CHANGE verb.

Consider the command:

```
CHANGE WHERE CLASS=MISSIL AND SIDE=BLUE
AND CATCODE=2763&2764&2765 SETTING ISITE=1&2&3
```

Note that in this command both the WHERE and SETTING clause contains an extended equals phrase. In this case, DATA will assume that there is a one-to-one correspondence between the records which satisfy each element of the extension in the WHERE clause and the change specified in the same position in the SETTING clause. In general, every record type combination which satisfies the WHERE clause will be changed to match the values given in the setting clause. In the example, the MISSIL whose CATCODE equals 2764 would have its ISITE set to 2.

It should also be noted that it is a simple matter for DATA to handle a CHANGE verb whose WHERE clause contains only an equals clause (extended or not) and the attribute is DESIG.

A helpful hint in the use of CHANGE verbs is the knowledge that the COP processes all verbs sequentially. For a whole series of CHANGE verbs could be used where more than one of the verbs alters the same record type but setting different attribute values.

5.2.3 The DELETE Verb. The DELETE verb is used when the user wishes to eliminate a particular record of a certain type. The user should note that deletion of records from an IDS data base can have far reaching consequences. The DELETE verb has two legal adverbs: ONPRINTS and WHERE. In general, after determining the set of record types to be retrieved according to the WHERE clause, DATA retrieves these record types and deletes the type that is most subordinate in the data organization.

For example, the DELETE verb sample in the sample below will cause deletion of all refuel points in region 1.

```
DELETE WHERE CLASS=REFUEL&IREG=1
```

5.3 Output

5.3.1 Standard Output. The DATA module has no standard output.

5.3.2 Optional Output.

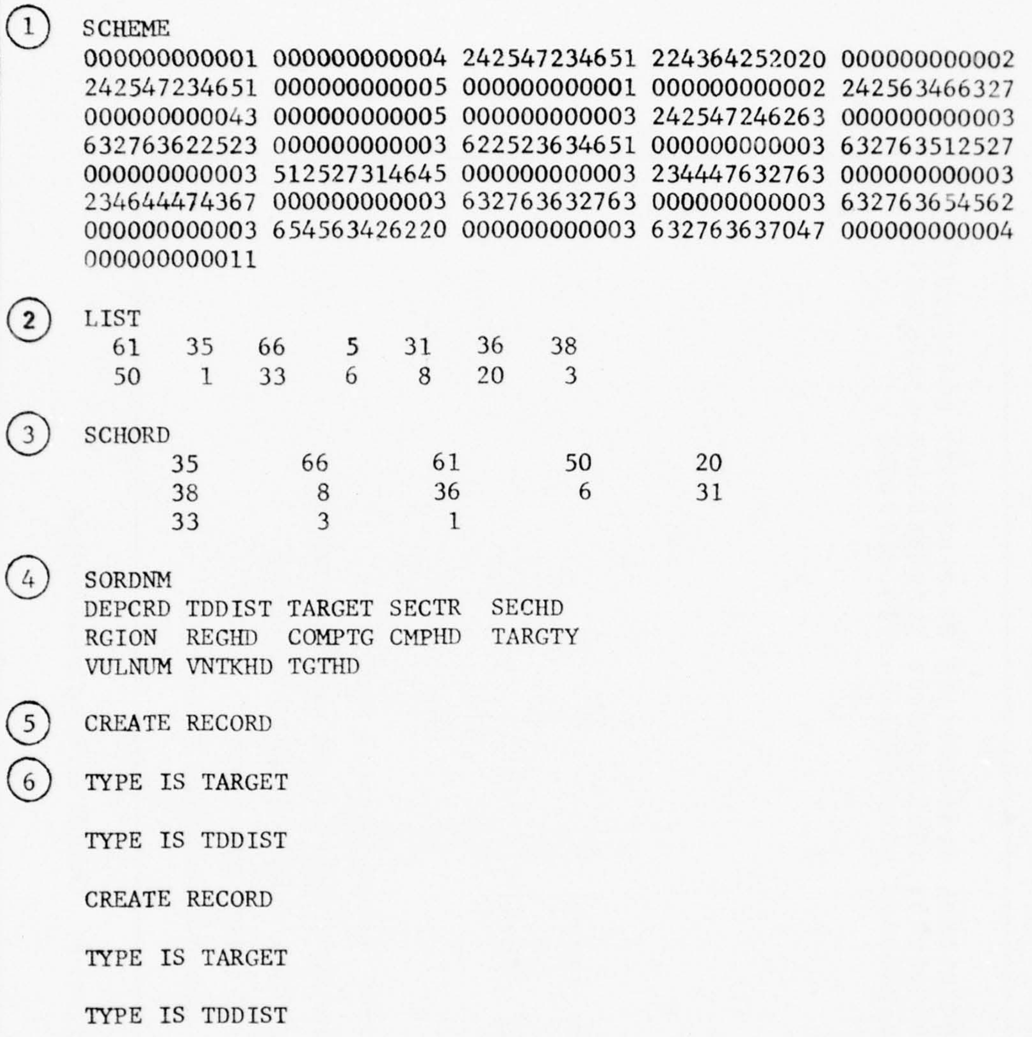
5.3.2.1 The CREATE Verb. The optional output of the CREATE verb consists of, for each supplied setting clause, a display of the retrieval scheme plus a message each time a record is created specifying its type. A sample of CREATE Verb output appears in figure 8.

5.3.2.2 The CHANGE Verb. The optional output of the CHANGE verb consists of a display of the retrieval scheme used plus a message each time a record is changed specifying its type. A sample appears in figure 9.

5.3.2.3 The DELETE Verb. The optional output of the DELETE Verb consists of a display of the retrieval scheme used plus a message each time a record is deleted specifying its type. A sample appears in figure 10.

5.3.2.4 Retrieval Scheme. The concept of data retrieval schemes (figures 8, 9, and 10) is identical for all verbs for this and other modules. Therefore, only one explanation is required and is as follows.

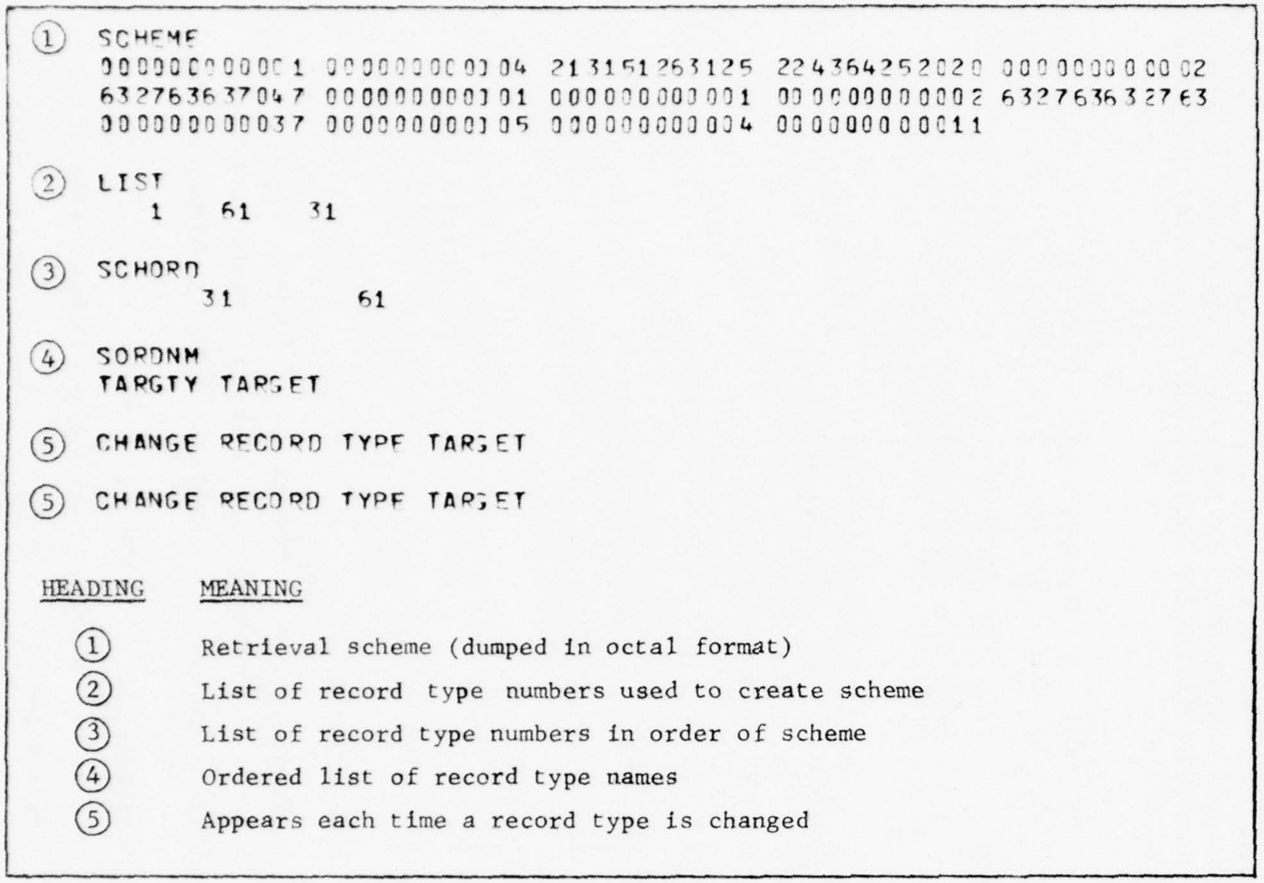
A COP retrieval scheme is a series of numbers which may be viewed as instructions. Each instruction consists of an introductory word which contains an identifying number, followed by one to three words which make up the remainder of the instruction. There are four instruction types: Get Header, Chain Next, Chain Master, and Return.



```
①  SCHEME
    000000000001 000000000004 242547234651 224364252020 000000000002
    242547234651 000000000005 000000000001 000000000002 242563466327
    000000000043 000000000005 000000000003 242547246263 000000000003
    632763622523 000000000003 622523634651 000000000003 632763512527
    000000000003 512527314645 000000000003 234447632763 000000000003
    234644474367 000000000003 632763632763 000000000003 632763654562
    000000000003 654563426220 000000000003 632763637047 000000000004
    000000000011

②  LIST
      61   35   66    5   31   36   38
      50    1   33    6    8   20    3

③  SCHORD
            35        66        61        50        20
            38         8        36         6        31
            33         3         1

④  SORDNM
    DEPCRD TDDIST TARGET SECTR  SECHD
    RGION  REGHD  COMPTG CMPHD  TARGTY
    VULNUM VNTKHD TGTHD

⑤  CREATE RECORD

⑥  TYPE IS TARGET

    TYPE IS TDDIST

    CREATE RECORD

    TYPE IS TARGET

    TYPE IS TDDIST
```

HEADING	MEANING
①	Retrieval scheme (dumped in octal format)
②	List of record type numbers used to create scheme
③	List of record type numbers in order in which they appear in scheme
④	Ordered list of record type names
⑤	Signal that a set of record types is being created
⑥	Appears for each type in set.

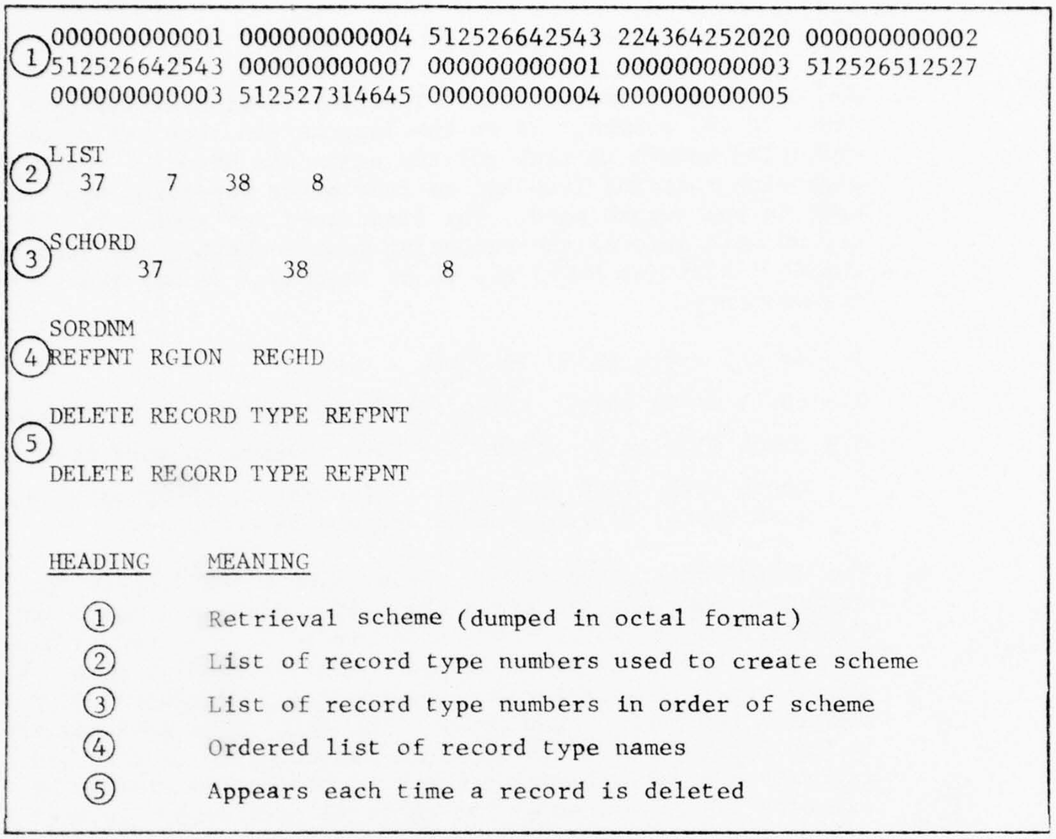
Figure 8. CREATE Verb Optional Output



```
① SCHEME
  000000000001 000000000104 213151263125 224364252020 000000000002
  632763637047 000000000101 000000000001 000000000002 632763632763
  000000000037 000000000105 000000000004 000000000011

② LIST
      1    61    31

③ SCHORD
          31          61

④ SORDNM
  TARGTY TARGET

⑤ CHANGE RECORD TYPE TARGET

⑤ CHANGE RECORD TYPE TARGET
```

HEADING	MEANING
①	Retrieval scheme (dumped in octal format)
②	List of record type numbers used to create scheme
③	List of record type numbers in order of scheme
④	Ordered list of record type names
⑤	Appears each time a record type is changed

Figure 9. CHANGE Verb Optional Output



```
  000000000001 000000000004 512526642543 224364252020 000000000002
① 512526642543 000000000007 000000000001 000000000003 512526512527
  000000000003 512527314645 000000000004 000000000005

  LIST
②   37     7    38     8

  SCHORD
③        37           38            8

  SORDNM
④ REFPNT  RGION   REGHD

  DELETE  RECORD  TYPE  REFPNT
⑤
  DELETE  RECORD  TYPE  REFPNT
```

HEADING	MEANING
①	Retrieval scheme (dumped in octal format)
②	List of record type numbers used to create scheme
③	List of record type numbers in order of scheme
④	Ordered list of record type names
⑤	Appears each time a record is deleted

Figure 10. DELETE Verb Optional Output

a. The Get Header Instruction -- This instruction always occurs first in the scheme. It is the instruction that tells the executing module to look for the next data header. This instruction contains from two to four words depending upon the code in the second word. The first word contains a 1. The second word informs the executing module whether the attribute CLASS or SIDE (or both) are to be checked. It has the following meanings:

   1 - do not check CLASS or SIDE

   2 - check CLASS only. Value for CLASS appears in word three.

   3 - check SIDE only. Value for SIDE appears in word three.

   4 - check both CLASS and SIDE. Value for CLASS appears in word three; value for SIDE appears in word four.

b. The Chain Next Instruction -- This instruction directs the executing module to retrieve the next record on a chain and informs it as to which instruction is to be executed if the master of the chain is retrieved. The instruction always has four words. The first word contains a 2. The second word contains the name of the chain. The third word contains the record type number of the chain's master. The fourth word contains a pointer (index number) to the instruction to be executed if the master of the chain is retrieved. In most cases, the pointer will be to the next instruction of the previous chain or the get header instruction.

c. The Chain Master Instruction -- This instruction directs the executing module to retrieve the master record of a chain. The instruction always has two words. The first word contains a 3. The second word contains the name of the chain.

d. The Return Instruction -- This instruction always appears last in a scheme. It informs the executing module that a logical set of records has been retrieved. It also informs the module of the instruction to execute next to retrieve the next logical set of records. The instruction contains two words. The first word contains a 4. The second word contains a pointer to a previous instruction.

5.3.2.5 Error Messages. Error messages for the CREATE, CHANGE, and DELETE verbs are shown in figures 11, 12, and 13.

1 SETTING CLAUSE WRONG FOR CREATE

An error has occurred in the setting clause, check inputs.

2 (F15.4) VIOLATES EDIT RANGE FOR (A12)

Edit error for floating point attribute. Value and attribute name are displayed.

3 (I10) VIOLATES EDIT RANGE FOR (A12)

Edit error for integer attribute: value and attribute name are displayed.

4 (A6) IS NOT IN EDIT LIST FOR (A12)

Edit error for alphabetic attribute: value and attribute name are displayed.

5 DATA//ERROR IN SAME AS CLAUSE

DATA could not find record identified in SAME clause.

6 ERROR IN TARGET INPUT

Error detected by Assignment table. Not sufficient amount of data supplied or the wrong data supplied.

7 LINKUP FAILURE

User supplied insufficient data to build a viable retrieval scheme.

Figure 11. CREATE Verb Error Messages

1 ERROR IN CHANGE VERB - MISSING CLAUSE

CHANGE verb requires both WHERE and SETTING clauses.

2 SETTING CLAUSE WRONG

Error in SETTING clause, check inputs.

3 WHERE CLAUSE WRONG

Error in WHERE clause, check inputs

4 NO RECORD TYPES DETERMINED IN CHANGE

User has not specified sufficient data to build a retrieval scheme.

5 LINKUP FAILURE

Same as above.

Figure 12. CHANGE Verb Error Messages

1 ERROR IN DELETE VERB

WHERE clause either in error or missing.

2 CANNOT BUILD SCHEME

User has not specified sufficient data to build a retrieval scheme.

Figure 13. DELETE Verb Error Messages

SECTION 6. EDITDB MODULE

6.1 General Purpose

The EDITDB module performs two similar but quite distinct functions. First, it performs the standard edit check as outlined in table 13. Second, it will perform any set of nonstandard edits desired by the user. This second function allows the user to check the validity of data attributes not covered by the standard edits or make sure that attributes fall within any desired limits other than those listed in the directory.

6.2 Input

The input to EDITDB consists of the single verb EDIT. If this verb appears without any clauses, then the standard edits (see table 13) will be used. If any clauses appear, EDITDB will perform nonstandard edits according to user specifications. The EDIT verb can have the following four adverbs: ONPRINTS, WHERE, FIELDS, and WITH. The ONPRINTS adverb stands alone and is used to generate non-standard prints.

6.2.1 WHERE Clause. This clause is used to specify a subset of the data base. Only those attributes within this subset are checked.

6.2.2 FIELDS Clause. The FIELDS clause has the general form:

FIELDS attribute [attribute . . . attribute]

Here the user specifies one or more attributes which are to be checked against either the range limits (numeric attributes) or list (alphabetic attributes) found in the data base directory.

6.2.3 WITH Clause. The WITH clause begins with WITH and is followed by a sequence of relational phrases each of which specifies the attribute (as subject) and the edit check. For example, the user may wish to check a numeric attribute as follows:

WITH CEP BETWEEN .01 and .5

or an alphabetic attribute as follows

WITH SIDE = BLUE OR RED

Any attribute may appear in both FIELDS and WITH clause in which case both the directory and user input criterion are checked separately.

6.3 Output

6.3.1 Standard Output. The EDITDB module has two sets of standard output. One appears in the case of a standard edit, an example of this appears in figure 14. Another is the nonstandard edit option which has standard output as shown in figure 15.

Table 13. EDITDB Standard Edits

1. ALRTDL Less than or Equal to NLRTDL

2. ALRTDB Less than or Equal to NLRTDB

3. ADBLI Less than or Equal to ADBLR

4. ATTRSU Less than or Equal to ATTRCO

5. FVALT1 Greater than or Equal to FVALT2
   FVALT2 Greater than or Equal to FVALT3
   FVALT3 Greater than or Equal to FVALT4
   FVALT4 Greater than or Equal to FVALT5

6. T1 Less than or Equal to T2
   T2 Less than or Equal to T3
   T3 Less than or Equal to T4
   T4 Less than or Equal to T5

7. NOALER Less than or Equal to NOINCO
   NOINCO Less than or Equal to NOPERSQ

8. RANGE Less than or Equal to RANGEREF

9. TARDEFLO Less than or Equal to TARDEFHI

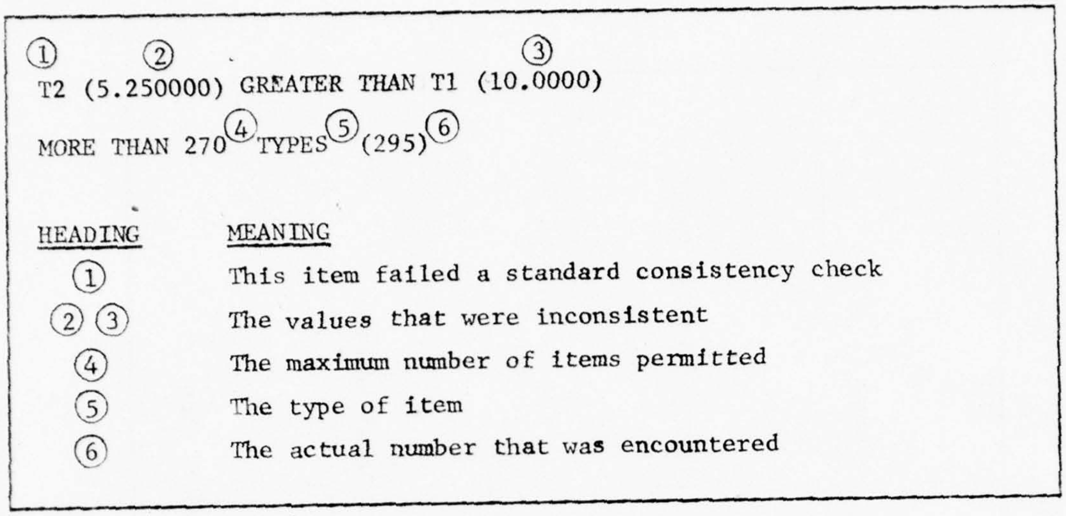
10. MINKILL Less than MAXKILL - .05

11. ALRTDL Less than NLRTDL - A
    where A = (NOALER/SIMLUN)*LCHINT
    Note: This is only checked if NOINCO is greater than NOALER

12. For every missile site
    NMPSIT =NOPERSQN*N
    Where N is the number of squadrons. N is calculated by identifying the lead squadron (ISITE > 0) and counting the number of squadrons such that (-ISITE = ISITE). This number is N-1.

13. Every depenetration corridor has a recovery base - MYRECOV1 = some legitimate DESIG.

14. The data base limits shown in Table 1 are checked.



```
①      ②                             ③
T2 (5.250000) GREATER THAN T1 (10.0000)

MORE THAN 270④TYPES⑤(295)⑥
```

HEADING	MEANING
①	This item failed a standard consistency check
② ③	The values that were inconsistent
④	The maximum number of items permitted
⑤	The type of item
⑥	The actual number that was encountered

Figure 14. EDITDB Standard Prints for Standard Edits

```
① CLASS IS BMBWEP SIDE IS BLUE WEPNAME IS B-52G
② FUNCTI ③ SBM IS NOT IN VALUE LIST ④
⑤   ICBM IRBM LRA MRBM SLBM TAC HIVAL LOWVAL

① CLASS IS MISSIL SIDE IS RED LAT IS 53.00000 LONG IS -50.00000 DESIG IS AB035
② LONG -50.00000 ③ NOT BETWEEN 0. ⑥ & 360.00000 ⑦

① CLASS IS COMPLX SIDE IS BLUE ICOMPL IS 3
  PHRASE NUMBER 5 ⑧ OF THE WITH CLAUSE WAS NOT MET

① CLASS IS AIRFIE SIDE IS RED TYPE IS STAGIN LAT IS 53.52000 LONG IS 310.0000 DESIG IS BA105
```

HEADING	MEANING
①	This line locates where the error occurred in the data base
②	The attribute that is in error
③	The value of the attribute
④	This message shows the value did not agree with a list of legal values
⑤	The list of legal values
⑥	The lowest legal value for a range comparison
⑦	The highest legal value for a range comparison
⑧	The 5th comparison in the WITH phrase was not valid for this item

Figure 15. EDITDB Standard Prints for Non-standard Edits

6.3.2 Optional Output. If the user includes the ONPRINTS adverb with a nonstandard run, the EDITDB module produces additional printed output. A sample of this appears in figure 16.

6.3.3 Error Message. The error messages produced by the EDITDB module only occur in the case of nonstandard edits. Figure 17 contains EDITDB module error messages.

```
① FIELDS DATA-NAMES,FORMAT,LOCATION,RANGE AND LIST
② LAT LONG FUNTI
③ F10.5  F10.5  A6
④ -90.00000    0.   8
⑤ 90.00000    360.00000   1
⑥ ICBM IRBM LRA MRBM SLBM TAC HIVAL LOWVAL
⑦ WITH CLAUSE POINTERS FOLLOWED BY THE CLAUSE
⑧ 1   48   69   97   108   123
⑨  000000000270 000000000000 000000000042 000000000012 014430000000
   000000000052 000000000004 000000000062 000000000006 000000000077
⑩ LOCATOR INFORMATION
⑪ FORMAT( '0 CLASS IS',A6,'SIDE IS',A6,'TYPE IS',A6
⑫ KEYAT =  1  2  59
⑬ NKEYAT = 3
```

HEADING	MEANING
①	This block of output is generated by a FIELDS clause
②	These attributes will be tested
③	These formats are used in printing the values of the attribute
④	The lowest value or a pointer to the list of legal values for the attribute
⑤	The highest value or a pointer to the list of legal values for the attribute
⑥	The list of values in 4 and 5
⑦	This block of output is generated by a WITH clause
⑧	An index of the starting point of phrases within the clause
⑨	The octal values of the WITH clause
⑩	The information used to locate an error in the data base
⑪	The format used in the first line of the error messages
⑫	The location of the attributes used to locate the error
⑬	The number of key attributes

Figure 16. EDITDB Optional Prints

1 $ (I5) IS NOT THE VERB NUMBER FOR EDIT

Will occur only if errors in the dictionary or Module link table cause erroneous call to EDITDB.

2 $ ILLEGAL ADVERB NUMBER (I5) - IT IS IGNORED

Will occur only if errors exist in syntax director.

3 $ ATFNDR UNABLE TO SET UP SCHEME

User has supplied insufficient data for module to build a viable retrieval scheme.

4 $ OF PHRASE IS MEANINGLESS (6I10)

FIELDS clause should not contain Of Phrases. Argument is print of input command.

5 $ SETFLD LOST (6I10)

Input FIELDS clause contains nonattribute items. Argument is input command.

6 $ FIELD IS NOT AN ATTRIBUTE IN C30 (6I10)

Attribute in FIELDS clause is not a data base attribute.

Figure 17. EDITDB Error Messages

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 7. REPORT MODULE

7.1 General Purpose

The purpose of the REPORT module is to give the user/analyst the capability to produce ad hoc print reports. REPORT is capable of addressing any set of attributes in the QUICK integrated data base and displaying them, plus any arithmetic calculations upon them, in virtually any format the user desires. REPORT is called by three verbs: DESIGN, PRINT, and ALTER.

The DESIGN verb is used to specify the format and the sort to be used to display the data, the calculated variables to be included and the subset of the data base to be used. Further the user may give the display a name so that it is retained in the data base.

The PRINT verb produces the display the user has designed. In addition, the user may specify a different data base subset with the PRINT verb.

The ALTER verb is used to make changes to an existing saved 'display' or to construct a new 'display' from a previous one.

7.2 Input

7.2.1 The DESIGN Verb. The DESIGN verb may have the following adverbs: DISPLAY, SETTING, SORT, WHERE, DEFINE, FORMAT, ONPRINTS. The general command form is:

```
DESIGN [DISPLAY display-name [{OLD}]]
                             [{NEW}]]

       [SETTING {PAGELENGTH} value]
                {LINELENGTH}
                {REPORTCODE}

       [SORT {variable } {ASCENDING } or {A}]
             {attribute} {DESCENDING}    {D}

       [WHERE where-clause]

       [DEFINE variable {  =   } value-expression]
                        {EQUALS}

       [FORMAT format-clause ]

       [ONPRINTS]
```

Section 3 detailed all of these adverbs. However major points concerning these adverbs will be highlighted within subsections to follow. Also within the discussion to follow (and for sections on the ALTER and PRINT verbs) reference will be continually made to a sample input as given in figure 18.

7.2.1.1 The DISPLAY Adverb. The DISPLAY clause is used to identify the display for future use. If it is omitted, the module will assign a general name. The clause consists of a display name of up to 12 characters and the option of either of the keywords OLD or NEW. If OLD appears, the module will replace the old display (stored within data base) with the new one. If NEW, the module will assure that no duplication of names occur. The default is NEW.

In the example (figure 18) the display name is set to TEST and is replacing an old display. All adverbs, then, that follow are 'owned' by the display name TEST.

7.2.1.2 The SETTING Adverb. The SETTING adverb as used with the DESIGN verb is used to specify non-default values for length of print page, line length, and print file unit. The only three attributes allowed are:

- PAGELENGTH - The length (or number of lines) of the physical page for output. (default=55 lines)

- LINELENGTH - The length (or number of characters) of the physical line for output. (default=120 characters)

- REPORTCODE - The print unit used for output.(default=42)

Referring to figure 18, defaults are used, except that only 15 lines are permitted per page.

7.2.1.3 The SORT Adverb. The SORT clause (see section 3) is used to specify the order in which records involved in the display are to be processed. The clause consists of one or more pairs of items. The first item is either an attribute found in the desired records or a variable calculated from the record. The second item of each pair specifies the order of the sort as either ASCENDING (or A) or DESCENDING (or D).

Figure 18 shows that the records to be printed will appear according to increasing range values.

7.2.1.4 The WHERE Adverb. This clause is used to subset the data base. The clause consists of a boolean statement whose elements are relational phrases which may refer to attributes or any normal defines (see below). It may not contain any attribute using the OF phrase.

The example informs REPORT that only those records on the BLUE side are of concern.

```
① ***** CENTRAL OPERATIONS PROCESSOR *****

② DESIGN DISPLAY TEST OLD
    SETTING PAGELENGTH=15
    SORT RANGE A
    WHERE SIDE=BLUE
    DEFINE SUMSPEED=SUMSPEED+SPEED
    DEFINE CEPREL=CEP*REL
    FORMAT HEADER '1' 100 X PAGENO
           TRAILER 30 X SUMSPEED IN F15.3
           LINE 3 X CLASS TYPE FUNCTI CEP REL
                CEPREL RANGE SPEED
③ PRINT DISPLAY TEST
④ ALTER DISPLAY TEST
     SETTING REPORTCODE=12
      SORT RANGE D
      FORMAT AFTER HEADER 1
             HEADER 'OLIST OF WEAPON TYPES'
             HEADER 150 SPACES
             TRAILER 150 SPACES
           REPLACE LINE 1
             LINE 3 X CLASS TYPE FUNCTI 150 X
                  'CEP=' CEP IN F4.2 'REL=' REL IN F4.2
                  150 X 'CEPREL=' CEPREL 'RANGE=' RANGE
                    'SPEED=' SPEED
③ PRINT DISPLAY TEST
```

HEADING	MEANING
①	COP input print header
②	DESIGN verb input sentence
③	PRINT verb input sentence
④	ALTER verb input sentence

Figure 18. REPORT Module Input Example

7.2.1.5 The DEFINE Adverb. The DEFINE clause is used to perform calculations upon attributes of the records in the data base so that the results may be part of the display. This is done through the creation of a dummy-attribute called a variable. The clause consists of a single EQUALS relational phrase in which the variable is named as the subject. The object of the phrase is any value expression involving attributes, previously defined variables or the new variable itself.

Section 3 details the various types of DEFINEs. Figure 18 shows two DEFINEs. One (CEPREL) will be printed within each print line; and the other (SUMSPEED) will accumulate data for each record processed and produce only a summary print. Note that the user only states needed attributes (SPEED, CEP, REL). Nothing is required of the user to inform REPORT where within the data base these attributes reside; the COP knows which record types contain these attributes and what subset to query based on the WHERE clause.

7.2.1.6 The FORMAT Adverb. This adverb is used to specify the format of each line to be printed (section 3 gives a detailed explanation of this clause). Basically, this adverb reasonably parallels the standard FORTRAN FORMAT statement where the variables to be printed, and print mode instructions are provided. Default modes of print (not IN word supplied) will be a format from the attribute's edit range in QUICK directory. These defaults are A7, I7, or F9.3.

The example shown (figure 18) prints variables: CLASS, TYPE, FUNCTION, CEP, REL, CEPREL RANGE, and SPEED. All are standard attributes except for the DEFINE variable CEPREL. Standard print formats will be obtained from the directory. The variable SUMSPEED is to be printed using F15.3 format, after 30 spaces have been skipped. Words HEADER and PAGENO are explained in section 3.

The example DISPLAY if printed as given via a PRINT verb (explained in following sections) will be as given in figure 19.

```
MSLWEP POL-A2 SLBM    1.1.    0.600 1500. 6000.
MSLWEP POL-A3 SLBM    1.1.    0.700 2500. 8000.
MSLWEP POSEID SLBM    1.1.    0.560 2500.10000.
TNKWEP KC-135 000000  0.0.    0.    2500.  470.
BMBWEP F-111  TAC     1.1.    0.450 3000.  500.
BMBWEP F-111B TAC     1.1.    0.450 3000.  500.
MSLWEP MM-IA  ICBM    1.1.    0.700 4700.12000.
BMBWEP B-58   LRA     1.1.    0.450 4800.  510.
MSLWEP MM-IB  ICBM    1.1.    0.600 5500.12000.
MSLWEP MM-II  ICBM    1.1.    0.474 6300.12000.
MSLWEP MM-III ICBM    1.1.    0.474 6300.12000.
BMBWEP B-52E  LRA     1.1.    0.450 6400.  485.
MSLWEP TITAN  ICBM    1.1.    0.560 7200.12000.
                                86465.000
```

Figure 19. Example of REPORT (Produced by DESIGN Verb Input Example)

7.2.2 The PRINT Verb. The DESIGN verb built instructions whereby a report could be developed. The PRINT verb simply informs REPORT to print the display by supplying a display name and any other subsetting information.

The PRINT verb may have three adverbs: DISPLAY, WHERE, and ONPRINTS.

The DISPLAY clause is used to identify the display to be printed. If it is omitted, the display that will be printed is that created by the last DESIGN verb with no DISPLAY clause.

The WHERE clause overrides any WHERE clause in the display to be printed and may contain any normal DEFINE contained in the display. However, since the retrieval scheme is not reconstructed, the WHERE clause should not contain any attribute which would not have been retrieved in the display prior to the addition of the new WHERE clause.

7.2.3 The ALTER Verb. The ALTER verb has one fixed adverb, DISPLAY, and six optional adverbs: SETTING, SORT, WHERE, DEFINE, FORMAT, and ONPRINTS. The general form of the ALTER verb is as follows:

```
ALTER  DISPLAY  display-name  [display-name]

           [SETTING  setting-values]

           [SORT  sort-clause]

           [WHERE  where-clause]

           [DEFINE  define-clause]

                    { REMOVE  }   { HEADER  }
           [FORMAT  { AFTER   }   { TRAILER }  numeric value]
                    { REPLACE }   { LINE    }
                                  { PAGE    }

                             [ { REMOVE  } ]
           format-phrase     [ { AFTER   } ]
                             [ { REPLACE } ]

       [ONPRINTS]
```

Explanations of each adverb follows plus reference is continually made to the example as given in figure 18.

7.2.3.1 The DISPLAY Adverb. This clause may contain either one or two display names. If one, the name is assumed to be that of an old display which the user wishes to alter or correct. If two names are present, the first is assumed to be that of a new display which the user wishes to create by altering the display named second.

7.2.3.2 The SETTING Adverb. This clause has the same meaning as in the DESIGN verb except that a parameter set in the old display remains as set unless changed in the new clause; the values do not revert to the defaults.

In the example (figure 18) REPORT CODE will be set to 12 and PAGELENGTH still equals 15.

7.2.3.3 The SORT and WHERE Adverbs. These clauses have the same meaning as in the DESIGN verb and replace any occurrence of the same adverb in the old display.

In the example (figure 18) the sort order is reversed from the original.

7.2.3.4 The DEFINE Adverb. The use of the DEFINE adverb is similar here to that with the DESIGN verb. Further, if the DEFINE variable name is a new one, it is added to the set of define variables. If the define variable name is the same as one in the old display, the new replaces the old. Finally, if a define variable is set equal only to itself it has the effect of eliminating any old variable of the same name.

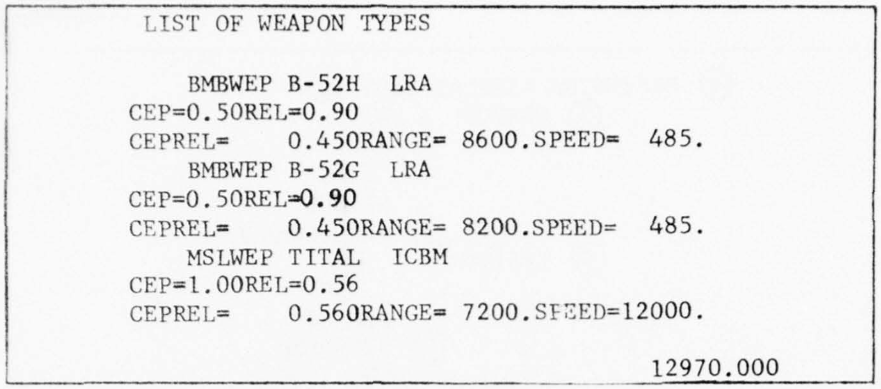
7.2.3.5 The FORMAT Adverb. The FORMAT clause of an ALTER verb contains three special words to direct the attention of an old FORMAT clause: AFTER, REMOVE and REPLACE. Each of these is followed by one of the special words: PAGE, HEADER, TRAILER or LINE. These special words are followed by an identifying number which can be found in the format definition report (see next section of this manual) of the display to be altered. The beginning of the format can be denoted by PAGE 0. The special words have the following effects:

- o AFTER - Any and all items which follow the identifier up to the next AFTER, REMOVE, or REPLACE are inserted after the identified item in the old format.

- o REMOVE - The identified item is removed from the format. (If the item named is PAGE every item up to the next PAGE is removed).

- o REPLACE - The identified item is removed and all items which follow the identifier up to the next AFTER, REMOVE or REPLACE are inserted in the removed items place. (If the item to be replaced is a PAGE item, the PAGE item alone is removed and no other action takes place.)

The ALTER verb (figure 18) changes the original output (figure 20) to that as given in figure 21.



```
        LIST OF WEAPON TYPES

            BMBWEP B-52H   LRA
      CEP=0.50REL=0.90
      CEPREL=     0.450RANGE= 8600.SPEED=  485.
            BMBWEP B-52G   LRA
      CEP=0.50REL=0.90
      CEPREL=     0.450RANGE= 8200.SPEED=  485.
            MSLWEP TITAL   ICBM
      CEP=1.00REL=0.56
      CEPREL=     0.560RANGE= 7200.SPEED=12000.

                                          12970.000
```

Figure 20. Example of REPORT (Produced by ALTER Verb Input Example)

7.3 Output

7.3.1 Standard Output. The standard output for the DESIGN verb appears in figure 21. The standard output for the ALTER verb appears in figure 22. Both DESIGN and ALTER verbs produce a format definition report which describes the format of the display and provides the numbering scheme used in future ALTER verbs. This report will not appear if the ALTER verb does not affect the format of the display. The PRINT verb has no standard output, however a sample report produced by the PRINT verb is shown in figures 20 and 21.

7.3.2 Optional Output. Both DESIGN and ALTER verbs cause the creation of a retrieval scheme. This scheme can be displayed as optional output and is explained in section 5.3.2.1. PRINT has no optional output.

7.3.3 Error Messages. REPORT module error messages appear in figure 23.

```
① DESIGNING DISPLAY TEST ②
       ③ HEADER  1 ④
                   1
                     100
                   SPACES
                ⑧ PAGE NUMBER
       ⑤ TRAILER   1 ④
                     30
                   SPACES
                ⑨ SUMSPEED
                   IN
                   F15.3   ⑩
       ⑥ LINE  1④
                      3
                   SPACES
                ⑦ ATTRIBUTE   1
                   ATTRIBUTE  51
                   ATTRIBUTE 128
                   ATTRIBUTE 122
                   ATTRIBUTE 144
                ⑨ CEPREL
                   ATTRIBUTE 125
                   ATTRIBUTE 123
```

HEADING	MEANING
①	Indicates that new display is being designed
②	New display name
③	Print of FORMAT clause: HEADER
④	Identifying number
⑤	Trailer
⑥	Line
⑦	Indicates Attribute is to be displayed - number is attribute number
⑧	Indicates page number is to be displayed
⑨	Indicates define variable is to be displayed
⑩	User input format.

Figure 21. DESIGN Standard Output

```
① NEW DISPLAY TEST ②       ALTERED FROM TEST ③
         ④ HEADER     1
                        1
                          100
                        SPACES
                        PAGE NUMBER
            HEADER    2
                        OLIST OF WEAPON TYPES
            HEADER    3
                          150
                        SPACES
            TRAILER   1
                           30
                        SPACES
                        SUMSPEED
                        IN
                        F15.3
            TRAILER   2
                          150
                        SPACES
            LINE      1
                            3
                        SPACES
                        ATTRIBUTE   1
                        ATTRIBUTE  51
                        ATTRIBUTE 128
                          150
                        SPACES
                        CEP=
                        ATTRIBUTE 122
                        IN
                        F4.2
                        REL=
                        ATTRIBUTE 144
                        IN
                        F4.2
                          150
                        SPACES
                        CEPREL=
                        CEPREL
                        RANGE=
                        ATTRIBUTE 125
                        SPEED=
                        ATTRIBUTE 123
```

Figure 22. ALTER Standard Output
(Part 1 of 2)

HEADING	MEANING
①	Indicates that new display is altered from old display (in this case old display is changed and is the same as new display)
②	Verb display name
③	Old display name
④	Print of FORMAT clause (see figure 18)

Figure 22. (Part 2 of 2)

1 ERROR-NO DISPLAY CLAUSE

Display clause missing from ALTER verb.

2 ERROR IN DISPLAY CLAUSE

Display clause input incorrectly.

3 OLD DISPLAY (A12) NOT FOUND

Argument is name of old display clause sought.

4 ERROR IN SETTING CLAUSE

Illegal attribute (only REPORTCODE, LINELENGTH and PAGELENGTH are allowed).

5 ERROR IN WHERE CLAUSE

WHERE clause entered incorrectly, check inputs.

6 ERROR IN DEFINE CLAUSE

DEFINE clause entered incorrectly.

7 ERROR IN FORMAT CLAUSE

Format clause entered incorrectly.

8 NEW DISPLAY (A12) ALREADY EXISTS

Argument is name of display which user tried to create but which already exists on file.

9 NO FORMAT CLAUSE

FORMAT clause left out (DESIGN only).

10 ERROR IN DEFINE CLAUSE (I3)

Analysis of DEFINE clause encountered error. Argument is occurrence of clause.

11 ERROR IN SORT CLAUSE

Order of argument is sort clause wrong or alphabetic is not a define name.

Figure 23. REPORT Module Error Messages
(Part 1 of 2)

12 DEFINES CANNOT BE RESOLVED

No order in which to execute DEFINEs can be found.

13 ILLEGAL DEFINE IN WHERE CLAUSE

DEFINE in WHERE clause is not a normal define.

14 ERROR IN INPUT WHERE CLAUSE

Error for WHERE clause input to PRINT verb.

15 ATFNDR UNABLE TO SET UP SCHEME

User provided insufficient data to set up viable retrieval scheme.

Figure 23. (Part 2 of 2)

SECTION 8. SAVE AND RESTORE MODULE (SRM)

8.1 General Purpose

The purpose of the SRM is to give the user the capability to copy the integrated data base onto tape and to restore the same data base to a previously stored state by reading in such a tape.

Technically, the entire QUICK system may be executed without using the SRM. Practically, this is not recommended.

8.2 Input

A SAVE command causes the COP to write the integrated data base onto a magnetic tape. There is no restriction as to when this command may be used. The general command form is:

```
SAVE [ON UNIT unit number]
```

The UNIT clause allows the user to change the output unit number which has a default value of 35.

A RESTORE command causes the COP to write the contents of a saved magnetic tape onto a disk file. The general form of the command is:

```
RESTORE [FROM UNIT unit number]
```

The UNIT clause allows the user to change the output unit number which has a default of 35.

8.3 Output

There is no standard output produced by this module since only data transfer is involved.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 9. EXTERNAL INTERFACE MODULE (EIM)

9.1 General Purpose

The purpose of the EIM is to create output tapes/files which are designed to be input to external processors. EIM has two command verbs: BUILD and PLOTDATA.

The BUILD verb has the capability to create two standard tapes plus a generalized output capability similar to that of the REPORT module. The user specifies the output file to be developed by inserting the special word SIDAC, TABLE, or OTHER within the BUILD command. The SIDAC word causes the production of two JAD format data base assessment tapes. The TABLE word causes the production of a standard set of tables. The OTHER word provides the capability to produce generalized data files.

The PLOTDATA verb produces an output tape suitable for use on the CALCOMP plotter. The plotable information includes penetration and depenetration corridors, refuel points and recovery bases.

9.2 Input

9.2.1 The BUILD Verb. The BUILD verb has six optional adverbs: UNIT, WHERE, SORT, DEFINE, FORMAT, and ONPRINTS. It also has one required adverb FILE. The general form is:

```
                  ⎧ TABLE ⎫
BUILD FILE        ⎨ SIDAC ⎬
                  ⎩ OTHER ⎭

. . . [ON UNIT numeric-value]

. . . [WHERE where-clause] . . . [SORT sort-clause]

. . . [DEFINE define-clause]

. . . [FORMAT format-clause]

. . . [ONPRINTS]
```

9.2.1.1 FILE TABLE. This clause may be accompanied by a UNIT, WHERE, and/or ONPRINTS clause. The standard output unit defaults to 35.

9.2.1.2 FILE SIDAC. This clause stands alone except for the optional ONPRINTS clause. BLUE targets are defined on unit 35 and RED targets on unit 36.

9.2.1.3 FILE OTHER. Generalized data files are produced through user inputs. A FORMAT clause must be used and UNIT, WHERE, SORT, DEFINE, ONPRINTS may be included.

9.2.1.4 The UNIT Adverb. The UNIT clause is used to specify an output unit other than 35.

9.2.1.5 The WHERE Adverb. For FILE TABLE or SIDAC instructions, this clause may only be used to specify a value for the SIDE attribute. If omitted the value of SIDE defaults to BLUE.

For FILE OTHER this clause is used to specify the subset of the data base to be accessed to build the output file. This clause is fully detailed in section 3 and examples of usage within the REPORT module, which is similar to the EIM, are presented in section 8. Normal DEFINE variables may be used. However, the OF phrase is not permitted.

9.2.1.6 SORT and DEFINE Adverbs. The use of these clauses are as given for the REPORT modules (section 8).

9.2.1.7 The FORMAT Adverb. The use of the FORMAT clause is different in the EIM from REPORT (section 8) in that headers, trailers and PAGNO must not be used. Furthermore the PAGE special word has a different meaning than explained elsewhere. When it encounters the word PAGE, the EIM writes an end-of-file mark on the output unit. (The user should declare the output tape as a multi-file.)

9.2.2 The PLOTDATA Verb. The PLOTDATA verb has one adverb SETTING. This adverb is followed by a series of equel relational phrases. Each phrase sets the values for a particular portion of the plot information. The general form of the PLOTDATA verb is as follows:

$$\underline{\text{PLOTDATA}} \quad \underline{\text{SETTING}}$$

$$\ldots \quad \underline{\text{MAP}} \left\{ \begin{array}{c} \underline{\text{EQUAL}} \\ \underline{=} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{PIC1}} \\ \underline{\text{PIC2}} \\ \underline{\text{PIC3}} \\ \underline{\text{PIC4}} \\ \underline{\text{PIC5}} \end{array} \right\}$$

$$\ldots \quad \underline{\text{SIDE}} \left\{ \begin{array}{c} \underline{\text{EQUAL}} \\ \underline{=} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{side}} \\ \underline{\text{BOTH}} \end{array} \right\} [\underline{\text{AND}}\ \underline{\text{side}}]$$

$$\ldots \quad \underline{\text{SCALE}} \left\{ \begin{array}{c} \underline{\text{EQUAL}} \\ \underline{=} \end{array} \right\} \underline{\text{numeric-value}}$$

$$\ldots \quad \underline{(}\underline{\text{LAT}},\ \underline{\text{LONG}}\underline{)} \left\{ \begin{array}{c} \underline{\text{EQUAL}} \\ \underline{=} \end{array} \right\} \underline{(}\underline{\text{latitude}},\ \underline{\text{longitude}}\underline{)}$$

$$\ldots \quad \underline{\text{(PLOT, LABEL)}} \left\{ \begin{matrix} \underline{\text{EQUAL}} \\ \underline{=} \end{matrix} \right\} \left(\left\{ \begin{matrix} \underline{\text{PENCOR}} \\ \underline{\text{DEPCOR}} \\ \underline{\text{RECOV}} \\ \underline{\text{REFUEL}} \end{matrix} \right\} \underline{,} \begin{matrix} \underline{\text{YES}} \\ \underline{\text{NO}} \end{matrix} \right) [\underline{\text{AND}} \ldots]$$

9.2.2.1 The MAP Phrase. This is used to set the type of plot. The user selects one of the following:

- o PIC1 - a PIC-1 (western hemisphere) plot
- o PIC2 - a PIC-2 (eastern hemisphere) plot
- o PIC3 - a 50 x 40 plot
- o PIC4 - a 20 x 40 plot
- o PIC5 - a 10 x 10 plot

9.2.2.2 The SIDE Phrase. This phrase selects the side or sides to be plotted. If the user wishes only one side plotted he states the side. If he wishes both sides plotted on the same plot he uses the value BOTH. If he wishes the sides plotted separately he uses an AND connector.

EXAMPLE: SIDE=BLUE AND RED

9.2.2.3 The SCALE Phrase. This phrase allows the user to set the ratio of real world units to map units.

EXAMPLE: SCALE=19010000

9.2.2.4 The Origin Setting Phrase. This phrase consists of the assignment of values to the attributes LAT and LONG. These values are to be used as the origin of the plot.

EXAMPLE: (LAT,LONG)=15.0,10.0)

9.2.2.5 The Plot Selection Phrase. This phrase details which data elements are to be plotted plus whether or not those elements are to be labeled. The equal phrase may be continued via the AND connector if multiple element types are desired.

EXAMPLE: (PLOT,LABEL)=(PENCOR,YES)AND(RECOV,NO)

9.3 Output

9.3.1 Standard Output

9.3.1.1 BUILD FILE SIDAC. Two tapes/files are generated; one for BLUE targets and the other for RED targets. There is no standard print.

9.3.1.2 BUILD FILE TABLE. The output consists of a tape/file on user direct unit. There is no standard output.

9.3.1.3 BUILD FILE OTHER. The output consists of a user defined output file. There is no standard print.

9.3.1.4 PLOTDATA. The output consists of a tape suitable for the CALCOMP plotter. There is no standard print.

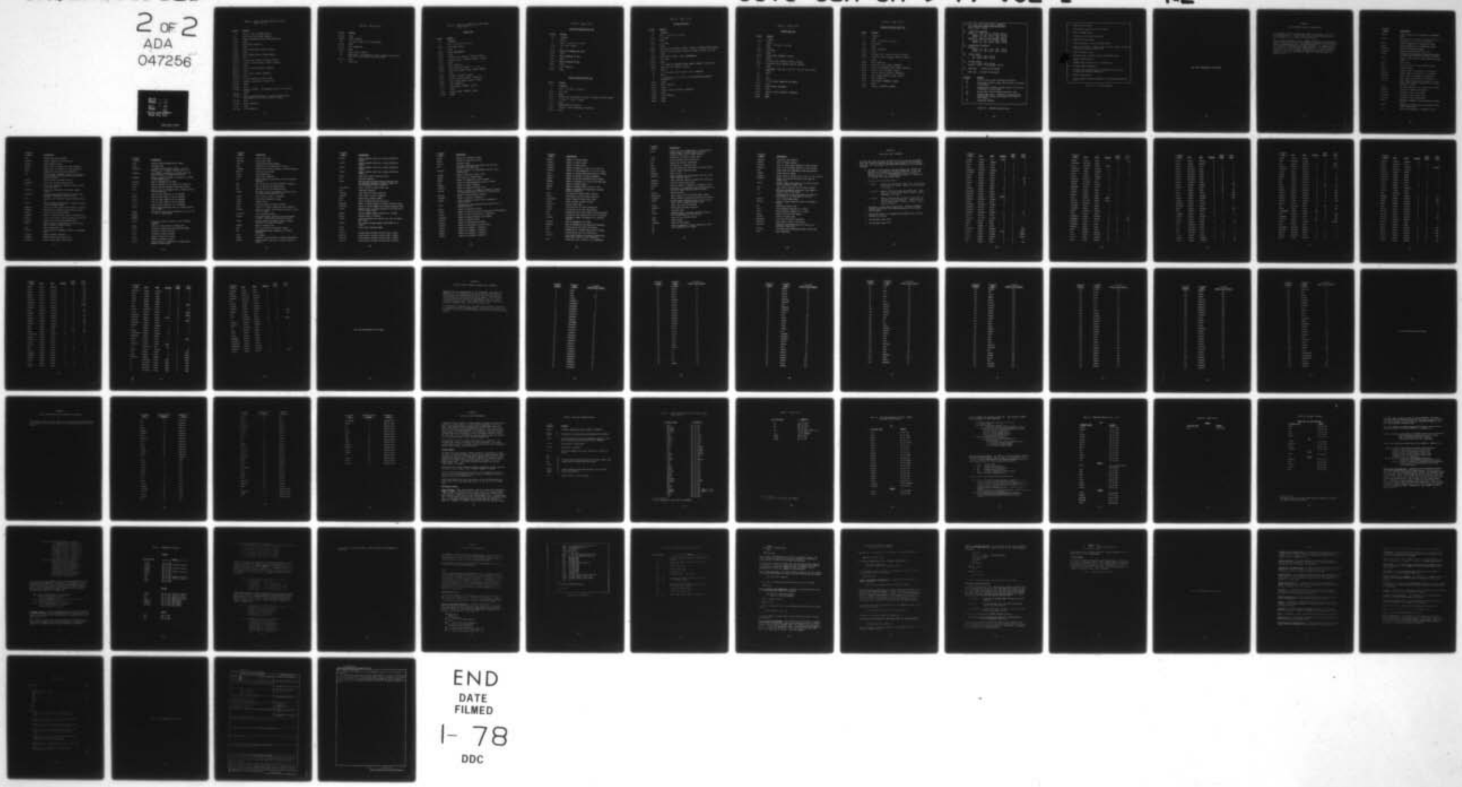
9.3.2 Optional Output. Print of records within files SIDAC, TABLE, and PLOTDATA can be obtained; table 14 gives the format of records within SIDAC; table 15 formats the TABLE records; figure 24 the PLOTDATA records, and figure 25 the error messages.

The optional output for BUILD FILE OTHER consists of a print of the retrieval scheme constructed during processing (see subsection 5.3.2.3).

AD-A047 256 SYSTEM SCIENCES INC BETHESDA MD F/G 15/7
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM. (QUICK). USE--ETC(U)
JUN 77 D J SANDERS, P F MAYKRANTZ, J M HERRIN DCA100-75-C-0019

UNCLASSIFIED CCTC-CSM-UM-9-77-VOL-1 NL

2 OF 2
ADA
047256



END
DATE
FILMED
1-78
DDC

Table 14. BUILD FILE SIDAC Output File Format
(Part 1 of 2)

Column	Meaning
1-5	Category code, (CATCODE) numeric
6-9	World Area Code (WACNO) alphabetic
10-15	Bomber Encyclopedia Number (BENO) alphabetic
16-20	Blank
21-26	Name (NAME) alphabetic
27-58	Blank
59-64	Major Complex Number (MAJOR) numeric
65-88	Blank
89-94	Minor Compound Number (MINOR) numeric
95-118	Blank
119-125	Latitude (LAT) degrees, minutes, seconds
126-133	Longitude (LONG) degrees, minutes, seconds
134-137	Blank
138-139	Country Location (CNTRYL) alphabetic
140-147	Blank
148-149	County Owner (CNTRYO) alphabetic
150-155	Blank
156-159	Severe vulnerability (VULN1) VNTK
160-163	Moderate vulnerability (VULN2) VNTK
164-167	"03P0"
168-190	Blank
191-198	Capacity (POP*10). This quantity is zero for all non-U/I targets
199-205	Blank
206-208	Radius (RADIUS*10) numeric - tenth of nautical miles. This quantity is zero for all non-U/I targets
209-288	Blank
289-293	DESIG, alphabetic
294	Blank
295-300	TYPE, alphabetic

Table 14. (Part 2 of 2)

Column	Meaning
301-303	Blank
304-305	ICLASS, numeric
306	1 for Blue targets; 2 for Red targets
307-312	Blank
313-314	TASK, alphabetic
315-318	Blank
319	Region (IREG), numeric
320	SAGA region. This quantity is IREG +1 unless country location is US or AK in which case it is IREG
321-335	Blank
336	Record Mark

Table 15. BUILD FILE TABLE Output File Formats
(Part 1 of 5)

TARGET LIST

Column	Meaning
1-8	'F1TARGET'
9	Side: 1 for Blue; 2 for Red
10-14	Line count, numeric
15	Blank
16-20	DESIG, alphanumeric
21-24	Blank
25-31	Latitude (LAT), degrees, minutes, seconds
32-39	Longitude (LONG), degrees, minutes, seconds
40	Blank
41-46	NAME, alphabetic
47-50	World Area Code (WACNO), alphabetic
51-55	Bomber Encyclopedia Number (BENO), numeric
56	Blank
57-61	Category (CATCODE), numeric
62-63	Country Location (CNTRYL), alphabetic
64-69	Major Complex Number (MAJOR), numeric
70-71	TASK, alphabetic
72-76	Index Number (INDEXNO), numeric
77	Blank
78-80	Complex Number (ICOMPL), numeric
81-90	Blank

Table 15. (Part 2 of 5)

VEHICLE CHARACTERISTICS LIST

Column	Meaning
1-7	'F1VEHIC'
8	Blank
9	Side: 1 for Blue; 2 for Red
10-14	Line count, numeric
15	'1'
16-20	Missile or bomber line count
21-55	Blank
56-58	CEP in hundreds of feet
59-61	Blank
62-64	CEP in hundreds of feet
65-69	Blank
70-75	TYPE, alphabetic
76-90	Blank

WEAPON CHARACTERISTICS LIST

Column	Meaning
1-8	'F1WEAPON'
9	Side: 1 for Blue; 2 for Red
10-14	Line count
15-17	Blank
18-19	Warhead type (Line count * 10 plus 1 for ASM; 0 for all others)
20	0 = Bomb, 1 = ASM, 2 = DECOY
21-37	Blank
38-43	Warhead yield in kilotons
44-46	Fission to Fusion percentage (FFRAC*100)
47-90	Blank

Table 15. (Part 3 of 5)

MISSILE BASE LIST

Column	Meaning
1-8	'F1MIBASE'
9	Side: 1 for Blue; 2 for Red
10-14	Line count
15	Blank
16-20	Line count
21	Blank
22-28	Latitude (LAT) degrees, minutes, seconds; S if South, blank if North
29-36	Longitude (LONG) degrees, minutes, seconds; E if East, W if West
37	Blank
38-41	Vulnerability Number (VULN1) alphanumeric
42-43	Type Count
44-45	Blank
46-47	'1/'; that is, beginning sortie number (always 1) followed by /
48-49	Number per site (NMPSIT), numeric
50	Blank
51	H if VN greater than or Equal to 20, S otherwise
52	Blank
53	1 if column 51 is H or if 51 is S and NOALER Equal NMPSIT Otherwise = 2
54-59	Blank
60-65	NAME, alphabetic
66-69	Blank
70-71	Country Location (CNTRYL), alphabetic
72-74	Blank
75-80	TYPE, alphabetic
81-84	Blank
85-90	BENO

Table 15. (Part 4 of 5)

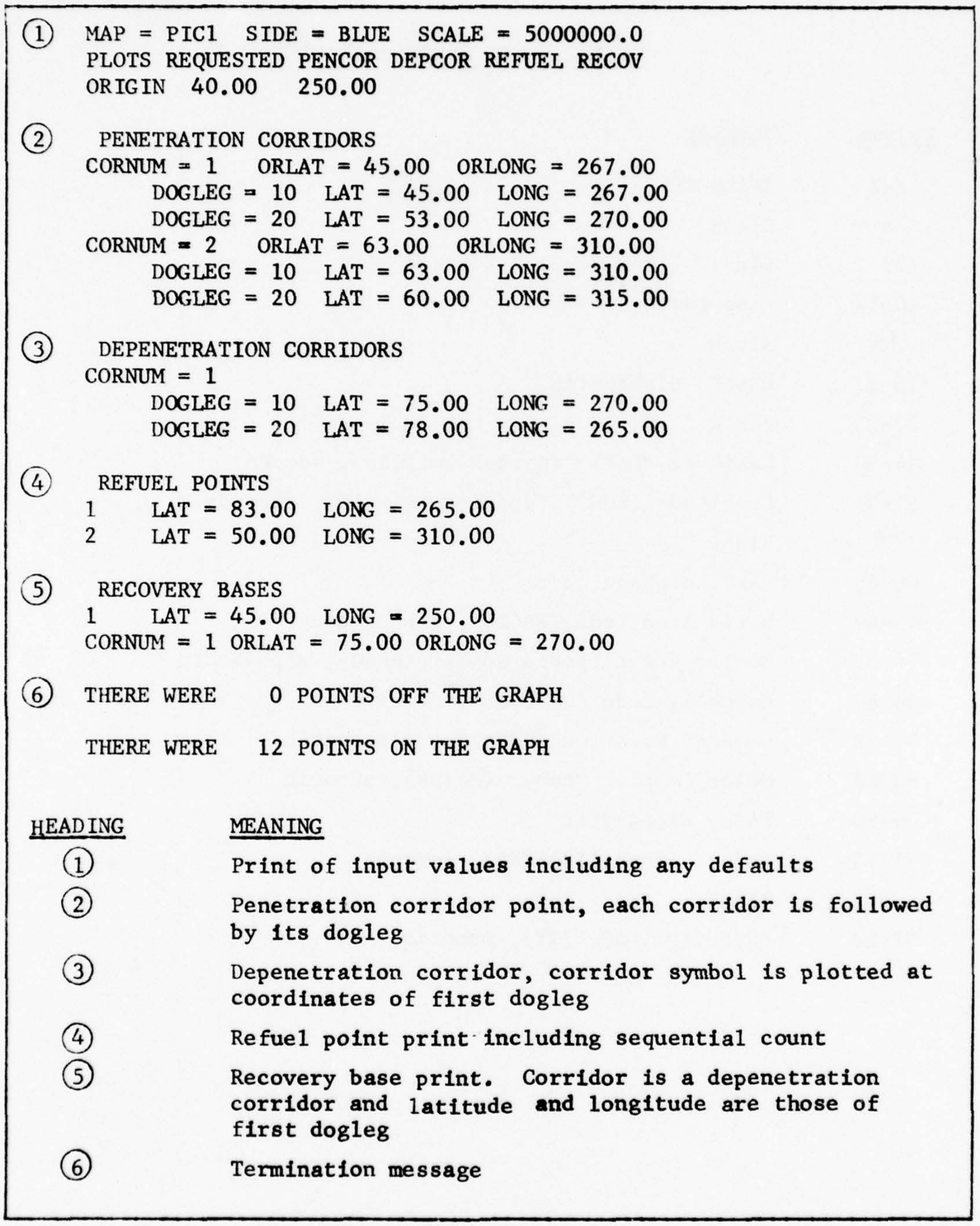
BOMBER BASE LIST

Column	Meaning
1-6	'F1BASE'
7-8	Blank
9	Side: 1 for Blue; 2 for Red
10	Blank
11-14	Line count
15	Blank
16-20	Index Number (INDEXNO), numeric
21	Blank
22-28	Latitude (LAT), degrees, minutes, seconds
29-36	Longitude (LONG), degrees, minutes, seconds
37	Blank
38	1 for SLBM, 2 for LRA, 3 for TAC, 7 for all others (from FUNCTI)
39	Blank
40	'X'
41-43	Blank
44	'X' for tanker, blank for all others
45-59	Blank
60-65	NAME of base, alphabetic
66-69	Blank
70-71	Country Location (CNTRYL), alphabetic
72-84	Blank
85-90	BENO

Table 15. (Part 5 of 5)

OFFENSIVE RECOVERY BASE LIST

Column	Meaning
1-7	'F1RECBS'
8	Blank
9	Side: 1 for Blue; 2 for Red
10-14	Line count
15	Blank
16-21	DESIG, alphabetic
22-23	Blank
24-30	Latitude (LAT), degrees, minutes, seconds
31-38	Longitude (LONG), degrees, minutes, seconds
39	Blank
40-45	NAME, alphabetic
46-49	World Area Code (WACNO), alphabetic
50-55	Bomber Encyclopedia Number (BENO), alphabetic
56-60	Category Code (CATCODE), numeric
61-62	Country Location (CNTRYL), alphabetic
63-68	Major Complex Number (MAJOR), numeric
69-70	TASK, alphabetic
71-75	Index Number (INDEXNO), numeric
76	Blank
77-80	Capacity (CAPACITY), numeric



```
①  MAP = PIC1  SIDE = BLUE  SCALE = 5000000.0
    PLOTS REQUESTED PENCOR DEPCOR REFUEL RECOV
    ORIGIN  40.00   250.00

②   PENETRATION CORRIDORS
    CORNUM = 1   ORLAT = 45.00  ORLONG = 267.00
         DOGLEG = 10   LAT = 45.00   LONG = 267.00
         DOGLEG = 20   LAT = 53.00   LONG = 270.00
    CORNUM = 2   ORLAT = 63.00  ORLONG = 310.00
         DOGLEG = 10   LAT = 63.00   LONG = 310.00
         DOGLEG = 20   LAT = 60.00   LONG = 315.00

③   DEPENETRATION CORRIDORS
    CORNUM = 1
         DOGLEG = 10   LAT = 75.00   LONG = 270.00
         DOGLEG = 20   LAT = 78.00   LONG = 265.00

④   REFUEL POINTS
    1    LAT = 83.00   LONG = 265.00
    2    LAT = 50.00   LONG = 310.00

⑤   RECOVERY BASES
    1    LAT = 45.00   LONG = 250.00
    CORNUM = 1 ORLAT = 75.00 ORLONG = 270.00

⑥  THERE WERE    0 POINTS OFF THE GRAPH

    THERE WERE   12 POINTS ON THE GRAPH
```

HEADING	MEANING
①	Print of input values including any defaults
②	Penetration corridor point, each corridor is followed by its dogleg
③	Depenetration corridor, corridor symbol is plotted at coordinates of first dogleg
④	Refuel point print including sequential count
⑤	Recovery base print. Corridor is a depenetration corridor and latitude and longitude are those of first dogleg
⑥	Termination message

Figure 24. PLOTDATA Optional Output

1 ERROR IN FILE CLAUSE

FILE clause contains error or is absent

2 ERROR IN WHERE CLAUSE

Attribute is not SIDE (TABLE only)

3 ERROR IN SETTING CLAUSE

Illegal attribute or value (PLOTDATA only)

POINT (I3) OFF MAP X = (F10.3) Y-10 = (F10.3) - point coordinates exceed plot limits (PLOTDATA Only)

4 ERROR IN FORMAT CLAUSE

Probable cause is illegal special word (OTHER only)

5 ERROR IN DEFINE CLAUSE

DEFINE clause contains an error (OTHER only)

6 DEFINES CANNOT BE RESOLVED

No order can be found in which to properly execute define variable calculations (OTHER only)

7 ERROR IN SORT CLAUSE

Sort parameters illegal, missing or in the wrong order (OTHER only)

Figure 25. EIM Error Messages

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

QUICK ATTRIBUTE NAMES AND DESCRIPTIONS

This appendix lists, in alphabetical order, the attributes used in the CCTC QUICK data base. Also provided are the definition/description of each attribute as it pertains to the QUICK system.

There are three attribute groups: gaming, non-gaming, and text English inputs. Gaming attributes are those elements necessary to define targets, weapons, and geographic data. Non-gaming elements are used to assist in the definition of Integrated Data Base structure. Text English attributes are part of the user input command structure. In this appendix an asterisk (*) after the attribute name implies a non-gaming attribute; a pound sign (#) implies text English inputs; all other attributes are gaming related.

ATTRIBUTE NAME	DESCRIPTION
ACLASS*	Class to which type is assigned by Assignment Table
ACTIVE	Value which is set true if a target class is to be processed by the allocation subsystem
ADBLI	ALRTDB probability for initiative attack
ADBLR	ALRTDB probability for retaliatory attack
ADVERBVL*	Adverb's identifying number
ALPLSTVL*	Value in attribute edit check list
ALRTDB	Probability of destruction before launch (DBL) of alert delivery vehicle (missile or bomber)
ALRTDL	Delay of alert vehicle before commencing launch (hours)
ALTDLY	Alert delay
ARRIVE	Time of arrival of fixed assigned weapon
ASIDE	Attacking side
ASNTASK*	Default value for attribute (in Directory)
ATDEFALT*	Default value for attribute (in Directory)
ATRNGLOW*	Lower limit for attribute (in Directory)
ATRNGHI*	Upper limit for attribute (in Directory)
ATTRBTYP*	Value which describes an attribute as to mode - alpha, list, integer, float; and type - single, multiple, control
ATTRCO	Attrition parameter for a penetration corridor
ATTRIBAD*	Attribute's common block address
ATTRIBN1*	Attribute's name (first half)
ATTRIBN2*	Attribute's name (second half)
ATTRIBNO*	Attribute's identifying number
ATTRLE	Attrition parameter for a penetration corridor dogleg
ATTRSU	Penetration corridor attrition parameter after defense suppression
ATYPE*	Legal type assignment in Assignment Table

ATTRIBUTE NAME	DESCRIPTION
BENO	Bomber Encyclopedia Number
CAPACITY	Vehicle capacity of recovery base
CATCODE	Target category code
CATHI*	Upper limit of category for type assignment
CATLO*	Lower limit of category for type assignment
CCREL	Command and control region reliability
CEP	Circular Error Probable, delivery error applicable to bomber and missile delivered weapons (nautical miles)
CEPASM	CEP for an air-to-surface missile
CHAINNAM*	Chain name in data structure index
CLASS	Class name assigned to identify various subsets of the data base
CLAUSESW*	Switch that identifies verbs which require clauses
CLAUSETY*	Value that identifies an adverb's clause type (Boolean, Sequential, Single or Null)
CMISS	Constant used in missile time-of-flight calculation
CNFLG*	Flag specifying country owner or location restriction in Assignment Table
CNTRYLOC	Country code for country where target is located
CNTRYOWN	Country code for country which owns the target
CORNUM	Corridor identification number
CORBOMB#	Number of nautical miles prior to corridor entry
CORMSL#	Fraction of missile flight completed at time zero
COUNTRY*	Legal country code in Assignment Table
DEFRAN	Typical range of interceptors at defense bases near a corridor (nautical miles)
DESIG	Target designator code
DESIGA2*	Alpha portion of assignable DESIG in Assignment Table
DISTANCE	General distance attribute
DISPNAM1*	REPORT display name (first half)
DISPNAM2*	REPORT display name (second half)

ATTRIBUTE NAME	DESCRIPTION
DOGLEG	Corridor dogleg identification number
DSIDE	Defending side
ELEMNTTY*	For adverbs with element phrases - type of element (e.g., operator, special word, etc.)
ELEMNTVL*	For ELEMNTTY of operator or special word, the exact operator or special word allowed
EXNBOMB#	Number of vehicle 'loads' of weapons to be added to each bomber group
EXNMIRV#	Same as EXNBOMB but for MIRVs
EXNMISS#	Same as EXNBOMB but for single shot missiles
FFRAC	Fission fraction (fission yield/total yield)
FLAG	Numeric code (1 through 9) used to impose allocation restrictions
FULL1*	Flag set when KOUNT1 is at its maximum
FULL2*	Flag set when KOUNT2 is at its maximum
FULL3*	Flag set when KOUNT3 is at its maximum
FULL4*	Flag set when KOUNT4 is at its maximum
FULL5*	Flag set when KOUNT5 is at its maximum
FUNCTI	Operational application code for a weapon system (e.g., ICBM)
FVALT1 FVALT2 FVALT3 FVALT4 FVALT5	Fraction of target value remaining at T1, T2, T3, T4, and T5, respectively
FVULN1	Fraction of value of target in first hardness component
GBASE	Number of launch bases in weapon group
GFRASM	Fraction of weapons in group which are ASMs
GLAT	Latitude of group centroid
GLONG	Longitude of group centroid
GNVEH	Number of vehicles in group
GNWPNS	Number of weapons in group
GPKNAV	Single shot kill probability of weapon group against a naval target

ATTRIBUTE NAME	DESCRIPTION
GREFCODE	Group refuel code
GREFTIME	Group refuel time
GROUP	Group identification number
GSBL	Probability of survival before launch
GSBLREAL	Same as GSBL but not adjusted for overallocation
GSTART	Starting weapon index
GTYPE	Group weapon type
GYIELD	Group yield (megatons)
HAZ	Lethal radius for air burst for first hardness component
HAZ2	Same as HAZ but for second hardness component
HGZ	Same as HAZ but for ground bursts
HGZ2	Same as HAZ2 but for ground bursts
HILOAT	The ratio of the low altitude attrition rate to the high altitude attrition rate
IALERT	Alert status; 1=alert 2=nonalert
ICLASS	Class index assigned for game
ICOMPL	Complex index
IDHOB	Preferred height of burst indicator
IGIW	Indices of General Industrial Worth (dollars)
INDEXNO	Index of a target used during processing to identify the target
INITSTRK#	Side initiating attack
IPENMO	Penetration mode; 1=aircraft uses penetration corridor, 0=penetration corridor not used
IRECMO	Recovery mode; 1=aircraft recovery planned, 0=aircraft recovery not planned
IREFUEL	Bomber refueling code
IREG	Index to identify a geographic region
IREP	Reprogramming index (capability of missile squadron)
ISITE	Site number
KORSTY	Parameter to adjust mode of corridor penetration
KOUNT1*	Highest numeric value for a unique DESIGA2 in region 1

ATTRIBUTE NAME	DESCRIPTION
KOUNT2*	Highest numeric value for a unique DESIGA2 in region 2
KOUNT3*	Highest numeric value for a unique DESIGA2 in region 3
KOUNT4*	Highest numeric value for a unique DESIGA2 in region 4
KOUNT5*	Highest numeric value for a unique DESIGA2 in region 5
LABEL#	Input; plot label selection switch
LAT	Latitude (degrees)
LCHINT	Time between successive vehicle launches from the same base (missile or bomber) subject to the simultaneous launch condition
LINELENGTH#	Input; length of report print line
LONG	Longitude (degrees)
LONGLOWR	Lower limit of sector longitude
LONGUPER	Upper limit of sector longitude
MAJOR	Target major complex number
MAP#	Input; PLOTDATA map type selection
MASDETNM*	Master or Detail Name in the data structure index
MASDETNO*	Master or Detail Number in the data structure index
MAXFRA	Maximum value of weapon resources to be used relative to target value
MAXKILL	Desired maximum damage expected for a target
MAXSAL	Maximum number of salvoes
MINCAP*	Lowest acceptable JAD capacity for type in Assignment Table
MINKILL	The required minimum damage established for a target
MINOR	Target minor compound number
MISDEF	
MYRECOV1	Depenetration corridor recovery base 1 (DESIG)
MYRECOV2	Depenetration corridor recovery base 2 (DESIG)
MYRECOV3	Depenetration corridor recovery base 3 (DESIG)
MYRECOV4	Depenetration corridor recovery base 4 (DESIG)

ATTRIBUTE NAME	DESCRIPTION
NADBLI	NLRTDB for initiative attack
NADBLR	NLRTDB for retaliatory attack
NALTDLY	Non-alert delay
NAME	Arbitrary alphanumeric descriptor for any item included in the data base
NAREADEC	Number of decoys per independent reentry vehicle for area BMD
NASMTYP	Number of distinct types of ASMs
NBASES	Total number of launch bases
NCCREG	Number of command and control regions
NCMS	Number of countermeasures carried by vehicle
NCOMPLX	Number of target complexes
NDCNTYCD	Number of distinct country codes
NDECOYS	Number of decoys on a bomber or number of decoys per independent reentry vehicle for terminal BMD
NDEPCRD	Number of depenetration corridors
NHRDCOMP	Number of hardness components
NLRTDB	Delay of nonalert vehicle before commencing launch (hours)
NLRTDL	Probability of destruction before launch (DBL) of nonalert vehicle
NMPSIT	Number of missiles per site
NOALER	Number of vehicles on alert status (also spelled NALERT)
NOINCO	Number of delivery vehicles in commission
NOPERSQN	Number of weapon vehicles per squadron
NPAYLOD	Number of payload tables
NPENCRD	Number of penetration corridors
NPNCRTY	Number of penetration corridor types
NPRSQ1	Number per squadron - scenario 1
NPRSQ2	Number per squadron - scenario 2
NPRSQ3	Number per squadron - scenario 3
NPRSQ4	Number per squadron - scenario 4

ATTRIBUTE NAME	DESCRIPTION
NRECOVB	Number of recovery bases
NREFUEL	Number of refuel points
NTANKERB	Number of tanker bases
NTARGETS	Number of targets as seen by the allocator
NTIMCOMP	Number of time components for a given target
NTINT	Number of terminal BMD interceptors at target
NUMASSGN	Number of weapons in fixed assignment
NUMDBL	Number of aircraft destroyed before launch
NUMLOAD	Number of warheads of a type in payload table
NWEPGRP	Number of weapon groups
NWEPTYP	Number of weapon types selected by user
NWHDS	Number of warheads per independent reentry vehicle (missiles)
ORLAT	Penetration corridor origin latitude
ORLONG	Penetration corridor origin longitude
PAGELENGTH#	Input; length of report print page
PAYALT	Bomber payload release altitude
PAYTBLNM	Payload table name
PCTIW#	Scaling factor for IGIW calculations
PCTPOP#	Scaling factor for POP calculations
PDES	Probability that launch failure destroys missile
PDUD	Probability that warhead will fail to detenate
PEXBOMB#	Fractional number of weapons to be added by PLANSET
PEXMIRV#	Same as PEXBOMB but for MIRVs
PEXMISS#	Same as PEXBOMB but for single shot missiles
PFIW#	Scaling factor for IGIW calculations
PFPF	Probability of failure during powered flight
PFPOP#	Scaling factor for POP calculations
PHRASETY*	Value that identifies an adverb's phrase type (Relational, Equal or Like, Elemental)
PINC	Probability that a missile is in commission

ATTRIBUTE NAME	DESCRIPTION
PKNAV	Single shot kill probability of a weapon against a naval target (a value greater than zero restricts weapon use to naval targets)
PLABT	Probability of vehicle launch abort
PLOT#	Input; PLOTDATA plot selection(s)
POP	Population (cities) (thousands)
PRABT	Probability of refueling abort
RADIUS	Size descriptor for area targets (nautical miles)
RANGE	Vehicle range (nautical miles)
RANGEASM	Range of ASM
RANGEDEC	Range decrement for low-altitude aircraft flight (high range/low range)
RANGEREF	Range (nautical miles) of bomber with refueling
RANGEMOD#	Adjusted weapon group range
REGION*	Region assigned to country in Assignment Table
REL	Reliability - probability that weapon system will arrive at target given successful launch
RELASM	ASM Reliability
REPORTCODE#	Input; report code for REPORT module output
RETARGET#	Input parameter specifying missiles may retarget
RNGMIN	Minimum range (nautical miles) for the missile type. Used in computing flight times
SCALE#	Input; map scale for PLOTDATA
SCENARIO#	DBMOD input scenario selection
SIDE	Item side name
SIMLUN	Maximum number of vehicle launches which can occur simultaneously from one base
SPDLO	Speed at low altitude (knots)
SPEED	Speed (knots)
SPEEDASM	Speed of ASM (knots)
T1 T2 T3 T4 T5	Times of departure of first through the fifth value components of a target

ATTRIBUTE NAME	DESCRIPTION
TABCHAR*	Dictionary tab character
TARDEF#	Level of target defense
TARDEFHI	Level of local bomber defense at high altitude
TABDEFLO	Level of local bomber defense at low altitude
TASK	Target task code indicating targeting priority
TGTMULT	Target multiple number
TGTNUMB	Target index in target list as given to the allocator
TGTREFCD	Target IDS internal Reference Code (used in target list)
TOFMIN	Minimum flight time (minutes) for missile types. Used in computing flight time
TTOS	Total time on station (for a tanker) (hours)
TYPE	Arbitrary alphameric designator (type name) to identify sets in data base
VAL	Relative value of an item within its CLASS as established in the data base by the user
VERBVAL*	Verb's identifying number
VONBASE	Number of launch base plus index of starting vehicle
VOZ	Normalized target value
VULN1	First hardness component of a target
VULN2	Second hardness component of a target
WACNO	Target World Area Code Number
WEPNAME	Subset of weapon type
WORDSTR1*	First half of word in dictionary
WORDSTR2*	Second half of word in dictionary
WORDTYPE*	Identifies dictionary word as to type (i.e., Attribute, Verb, etc.)
WORDVAL*	Dictionary word identifying number within type
YIELD	Yield (Negatons)

APPENDIX B

QUICK DATA BASE DIRECTORY

The QUICK data base directory consists of a list of all the attributes which can be used to describe the data items defined in the integrated data base. The information contained in the directory for each attribute includes:

a. The name of the attribute plus an indicator that defines the attribute as belonging to a logical collection. If the name appears alone it is a gaming attribute; if an asterisk (*) follows the name it is a non-gaming attribute; if a pound sign (#) follows it is text English input

b. The type of the attribute may be:

   o Single - appears only within one record type. Input values are either LIST, INTGER, ALPHA, or FLOAT depending on the mode.

   o Multiple - appears within more than one record type. Input values are either MLTLST, MLTINT, MLTALP, or MLTFLT depending on the mode.

   o Control - same as multiple plus is used as a match key for internal structure definition. Input values are either CNTLST, CNTINT, CNTALP, or CNTFLT depending on the mode.

c. The modes, or input/output conversions. These are standard FORTRAN formats plus a list which specifies a list of alphabetic entries.

d. The default value to be assigned the attribute when it is not defined for an item.

e. The attribute lower limit

f. The attribute upper limit

ATTRIBUTE NAME	TYPE	MODE	DEFAULT	LOWER LIMIT	UPPER LIMIT
ACLASS*	Single	List	OTHER	-	-
ACTIVE	Single	Integer	1	-	-
ADBLI	Single	Float	0	0	1
ADBLR	Single	Float	0	0	1
ADVERBVL*	Control	Integer	0	-	-
ALPLSTVL*	Single	Alpha	-	-	-
ALRTDB	Single	Float	0	0	1
ALRTDL	Single	Float	0	0	168
ALTDLY	Single	Float	0	0	99
ARRIVE	Single	Float	0	0	99
ASIDE	Single	Alpha	-	-	-
ASNTASK*	Single	Alpha	ZZ	-	-
ATDEFALT*	Single	Alpha	-	-	-
ATRNGLOW*	Single	Alpha	0	-	-
ATRNGHI*	Single	Alpha	999999	-	-
ATTRBTYP*	Multiple	Integer	0		
ATTRCO	Single	Float	0	0	1
ATTRIBAD*	Multiple	Integer	0	-	-
ATTRIBN1*	Control	Alpha	-	-	-
ATTRIBN2*	Control	Alpha	-	-	-
ATTRIBNO*	Multiple	Integer	0	0	241
ATTRLE	Single	Float	0	0	1
ATTRBTYP*	Multiple	Integer	0	-	-
ATTRSU	Single	Float	0	0	1
ATYPE*	Single	Alpha	-	-	-
BENO	Single	Alpha	-	-	-
CAPACITY	Single	Integer	0	0	9999
CATCODE	Single	Integer	75100	1	99999
CATHI*	Single	Integer	0	1	99999
CATLO*	Single	Integer	0	1	99999
CCREL	Single	Float	1	0	1
CEP	Single	Float	1	0	100

ATTRIBUTE				LOWER	UPPER
NAME	TYPE	MODE	DEFAULT	LIMIT	LIMIT
CEPASM	Single	Float	1	0	100
CHAINNAM*	Multiple	Alpha	-	-	-
CLASS	Multiple	Alpha	INDEX	-	-
CLAUSESW*	Single	Integer	0	0	1
CLAUSETY*	Single	Integer	2	1	4
CMISS	Single	Float	1	0	1
CNFLG*	Single	Integer	0	-	-
CNTRYLOC	Multiple	Alpha	-	-	-
CNTRYOWN	Multiple	Alpha	-	-	-
CORNUM	Multiple	Integer	1	-	-
CORBOMB#	Single	Float	0	-	-
CORMSL#	Single	Float	0	-	-
COUNTRY*	Single	Alpha	-	-	-
DEFRAN	Single	Float	0	-	-
DESIG	Control	Alpha	ZZ999	-	-
DESIGA2*	Multiple	Alpha	ZZ999	-	-
DISTANCE	Multiple	Float	0	-	-
DISPNAM1*	Single	Alpha	-	-	-
DISPNAM2*	Single	Alpha	-	-	-
DOGLEG	Multiple	Integer	1	-	-
DSIDE	Single	Alpha	-	-	-
ELEMNTTY*	Single	Integer	0	0	10
ELEMNTVL*	Single	Integer	0	0	999
EXNBOMB#	Single	Float	0	-	-
EXNMIRV#	Single	Float	0	-	-
EXNMISS#	Single	Float	0	-	-
FFRAC	Single	Float	1	0	1
FLAG	Multiple	Integer	0	0	9
FULL1*	Single	Integer	0	-	-
FULL2*	Single	Integer	0	-	-
FULL3*	Single	Integer	0	-	-

ATTRIBUTE				LOWER	UPPER
NAME	TYPE	MODE	DEFAULT	LIMIT	LIMIT
FULL4*	Single	Integer	0	-	-
FULL5*	Single	Integer	0	-	-
FUNCTI	Single	List	-	-	-
FVALT1	Multiple	Float	1	0	1
FVALT2	Multiple	Float	1	0	1
FVALT3	Multiple	Float	1	0	1
FVALT4	Multiple	Float	1	0	1
FVALT5	Multiple	Float	1	0	1
FVULN1	Multiple	Float	1	0	1
GBASE	Single	Integer	0	0	150
GFRASM	Single	Float	0	0	1
GLAT	Single	Float	0	-90	90
GLONG	Single	Float	0	0	360
GNVEH	Single	Integer	1	1	999
GNWPNS	Single	Integer	1	1	999
GPKNAV	Single	Float	0	-	-
GREFCODE	Single	Integer	1	-	-
GREFTIME	Single	Float	1	-	-
GROUP	Control	Integer	0	0	999
GSBL	Single	Float	0	-	1
GSBLREAL	Single	Float	0	-	1
GSTART	Single	Integer	0	-	-
GTYPE	Single	Alpha	-	-	-
GYIELD	Single	Float	0	0	99
HAZ	Multiple	Float	-	-	-
HAZ2	Multiple	Float	-	-	-
HGZ	Multiple	Float	-	-	-
HGZ2	Multiple	Float	-	-	--
HILOAT	Single	Float	.1	0	1
IALERT	Single	Integer	1	1	2
ICLASS	Single	Integer	1	1	15

ATTRIBUTE NAME	TYPE	MODE	DEFAULT	LOWER LIMIT	UPPER LIMIT
ICOMPL	Control	Integer	0	1	999
IDHOB	Multiple	Integer	-	0	1
IGTW	Single	Integer	0	0	99999
INDEXNO	Multiple	Integer	1	1	99999
INITSTRK#	Single	Integer	2	1	2
IPENMO	Single	Integer	1	0	1
IRECMO	Single	Integer	1	-1	1
IREFUEL	Single	Integer	0	-5	20
IREG	Control	Integer	0	0	20
IREP	Single	Integer	0	0	5
ISITE	Single	Integer	0	-100	100
KORSTY	Single	Integer	0	0	10
KOUNT1*	Single	Integer	0	0	999
KOUNT2*	Single	Integer	0	0	999
KOUNT3*	Single	Integer	0	0	999
KOUNT4*	Single	Integer	0	0	999
KOUNT5*	Single	Integer	0	0	999
LABEL#	Single	Alpha	-	-	-
LAT	Multiple	Float	0	-90	90
LCHINT	Single	Float	0	0	999
LINELENGTH#	Single	Integer	120	-	-
LONG	Multiple	Float	0	0	360
LONGLOWR	Single	Float	0	0	360
LONGUPER	Single	Float	360	0	360
MAJOR	Single	Integer	0	1	999999
MAP#	Single	Alpha	-	-	-
MASDETNM*	Multiple	Alpha	-	-	-
MASDETNO*	Multiple	Integer	-	-	-
MAXFRA	Multiple	Float	1	0	1
MAXKILL	Multiple	Float	1	0	1
MAXSAL	Single	Integer	0	0	24

ATTRIBUTE NAME	TYPE	MODE	DEFAULT	LOWER LIMIT	UPPER LIMIT
MINCAP*	Single	Integer	0	-	-
MINKILL	Multiple	Float	0	0	1
MINOR	Single	Integer	0	0	999999
MISDEF	Multiple	Integer	-	-	-
MYRECOV1	Single	Alpha	-	-	-
MYRECOV2	Single	Alpha	-	-	-
MYRECOV3	Single	Alpha	-	-	-
MYRECOV4	Single	Alpha	-	-	-
NADBLI	Single	Float	0	0	1
NADBLR	Single	Float	0	0	1
NALERT	Single	Integer	-	-	-
NALTDLY	Single	Float	-	-	-
NAME	Multiple	Alpha	-	-	-
NAREADEC	Single	Integer	0	0	50
NASMTYP	Single	Integer	0	0	999
NBASES	Single	Integer	0	0	999
NCCREG	Single	Integer	0	0	9
NCMS	Single	Integer	0	0	5
NCOMPLX	Single	Integer	0	0	-
NDCNTYCD	Single	Integer	0	0	9
NDECOYS	Single	Integer	0	0	10
NDEPCRD	Single	Integer	0	0	99
NHRDCOMP	Multiple	Integer	1	1	2
NLRTDB	Single	Float	-	-	-
NLRTDL	Single	Float	-	-	-
NMPSIT	Single	Integer	0	0	999
NOINCO	Single	Integer	0	0	999
NOPERSQ	Single	Integer	0	0	999
NPAYLOD	Single	Integer	0	0	999
NPENCRD	Single	Integer	0	0	99
NPNCRTY	Single	Integer	0	0	99

ATTRIBUTE				LOWER	UPPER
NAME	TYPE	MODE	DEFAULT	LIMIT	LIMIT
NPRSQ1	Single	Integer	0	0	999
NPRSQ2	Single	Integer	0	0	999
NPRSQ3	Single	Integer	0	0	999
NPRSQ4	Single	Integer	0	0	999
NRECOVB	Single	Integer	0	0	999
NREFUEL	Single	Integer	0	0	999
NTANKERB	Single	Integer	0	0	999
NTARGETS	Single	Integer	0	0	99999
NTIMCOMP	Single	Integer	1	1	5
NTINT	Multiple	Integer	0	0	999
NUMASSGN	Single	Integer	0	0	99
NUMDBL	Single	Integer	0	0	99
NUMLOAD	Single	Integer	0	0	99
NWEPGRP	Single	Integer	0	0	999
NWEPTYP	Single	Integer	0	0	999
NWHDS	Single	Integer	1	1	16
ORLAT	Single	Float	0	-90	90
ORLONG	Single	Float	0	0	360
PAGELENGTH#	Single	Integer	55	-	-
PAYALT	Single	List	-	-	-
PAYTBLNM	Single	Alpha	-	-	-
PCTIW#	Single	Float	-	-	-
PCTPOP#	Single	Float	-	-	-
PDES	Single	Float	0	0	1
PDUD	Single	Float	0	0	1
PEXBOMB#	Single	Float	-	-	-
PEXMIRV#	Single	Float	-	-	-
PEXMISS#	Single	Float	-	-	-
PFIW#	Single	Float	-	-	-
PFPF	Single	Float	0	0	1
PFPOP#	Single	Float	-	-	-

ATTRIBUTE NAME	TYPE	MODE	DEFAULT	LOWER LIMIT	UPPER LIMIT
PHRASETY*	Single	Integer	1	1	3
PINC	Single	Float	0	0	1
PKNAV	Single	Float	0	0	1
PLABT	Single	Float	0	0	1
PLOT#	Single	Alpha	-	-	-
POP	Single	Float	0	0	30000
PRABT	Single	Float	0	0	1
RADIUS	Multiple	Float	0	0	50
RANGE	Single	Float	0	0	20000
RANGEASM	Single	Float	0	0	20000
RANGEDEC	Single	Float	1.0001	0	10
RANGEREF	Single	Float	0	0	20000
RANGEMOD#	Single	Float	-	-	-
REGION*	Single	Integer	1	1	4
REL	Single	Float	1	0	1
RELASM	Single	Float	1	0	1
REPORTCODE#	Single	Integer	42	-	-
RETARGET#	Single	Alpha	-	-	-
RNGMIN	Single	Float	0	0	20000
SCALE#	Single	Float	-	-	-
SCENARIO#	Single	Alpha	INDIA	-	-
SIDE	Multiple	List	RED	-	-
SIMLUN	Single	Integer	1	1	50
SPDLO	Single	Float	0	0	10000
SPEED	Single	Float	0	0	30000
SPEEDASM	Single	Float	0	0	30000
T1	Multiple	Float	1000	0	1000
T2	Multiple	Float	1000	0	1000
T3	Multiple	Float	1000	0	1000
T4	Multiple	Float	1000	0	1000
T5	Multiple	Float	1000	0	1000

ATTRIBUTE				LOWER	UPPER
NAME	TYPE	MODE	DEFAULT	LIMIT	LIMIT
TABCHAR*	Single	Alpha	-	-	-
TARDEF#	Single	List	-	-	-
TARDEFHI	Multiple	Integer	0	0	7
TARDEFLO	Multiple	Integer	0	0	7
TASK	Multiple	Alpha	-	-	-
TGTMULT	Single	Integer	-	-	-
TGTNUMB	Single	Integer	-	-	-
TGTREFCD	Single	Integer	-	-	-
TOFMIN	Single	Float	0	0	1000
TTOS	Single	Float	0	0	100
TYPE	Multiple	Alpha	-	-	-
VAL	Multiple	Float	.01	0	99999
VERBVAL*	Control	Integer	-	-	-
VONBASE	Single	Integer	-	-	-
VOZ	Multiple	Float	-	-	-
VULN1	Control	Alpha	-	-	-
VULN2	Single	Alpha	-	-	-
WACNO	Single	Alpha	-	-	-
WEPNAME	Control	Alpha	-	-	-
WORDSTR1*	Single	Alpha	-	-	-
WORDSTR2*	Single	Alpha	-	-	-
WORDTYPE*	Single	Integer	-	-	-
WORDVAL*	Single	Integer	-	-	-
YIELD	Single	Float	0	0	9999

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

QUICKS ATTRIBUTE NUMBER AND NAMES AND ADDRESSES

Appendix A gave a total account of all attributes used within the QUICK system. This appendix repeats those attributes but ordered according to the attribute number which is used for internal processing and sometimes included within printed reports. Also included is the position of each gaming and non-gaming attribute as stored internally in common /30/. This common block is used to communicate between all module values stored for any record type.

If an asterisk (*) appears after the name, that attribute is defined as a non-gaming attribute; the appearance of a pound sign (#) implies the attribute is used for text English input; all other attributes are gaming related.

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
1	CLASS	1
2	SIDE	2
3	DESIG	130
4	ICLASS	3
5	ATTRIBN1*	5
6	ATTRIBN2*	6
7	ATTRIBAD*	7
8	ATTRIBNO*	8
9	ATTRBTYP*	9
10	ATDEFALT*	10
11	ATRNGLOW*	11
12	ATRNGHI*	12
13	CHAINNAM*	13
14	MASDETNM*	14
15	MASDETNO*	15
16	ALPLSTVL*	17
17	TABCHAR*	19
18	WORDSTR1*	21
19	WORDSTR2*	22
20	WORDTYPE*	23
21	WORDVAL*	24
22	VERBVAL*	25
23	CLAUSESW*	26
24	ADVERBVL*	27
25	CLAUSETY*	28
26	PHRASETY*	29
27	ELEMNTTY*	31
28	ELEMNTVL*	32
29	DISPNAM1*	33
30	DISPNAM2*	34

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
31	COUNTRY*	35
32	REGION*	36
33	CATLO*	37
34	CATHI*	38
35	CNFLG*	39
36	ASNTASK*	40
37	MINCAP*	41
38	ACLASS*	43
39	ATYPE*	45
40	DESIGA2*	47
41	KOUNT1*	48
42	KOUNT2*	49
43	KOUNT3*	50
44	KOUNT4*	51
45	KOUNT5*	52
46	FULL1*	53
47	FULL2*	54
48	FULL3*	55
49	FULL4*	56
50	FULL5*	57
51	TYPE	59
52	VULN1	60
53	VULN2	61
54	FVULN1	62
55	T1	63
56	T2	64
57	T3	65
58	T4	66
59	T5	67
60	FVALT1	68

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
61	FVALT2	69
62	FVALT3	70
63	FVALT4	71
64	FVALT5	72
65	CNTRYLOC	73
66	CNTRYOWN	74
67	FLAG	75
68	NHRDCOMP	76
69	NTIMCOMP	77
70	GROUP	79
71	GLAT	80
72	GLONG	81
73	GNWPNS	82
74	GNVEH	83
75	GTYPE	84
76	IALERT	85
77	GSBL	86
78	GSBLREAL	87
79	GREFCODE	88
80	GREFTIME	89
81	GYIELD	90
82	GFRASM	91
83	IREG	95
84	CORNUM	97
85	KORSTY	98
86	HILOAT	99
87	ATTRCO	100
88	ATTRSU	101
89	ORLAT	102
90	ORLONG	103

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
91	ICOMPL	109
92	LAT	110
93	LONG	111
94	HGZ	112
95	HGZ2	113
96	HAZ	114
97	HAZ2	115
98	MAXKILL	116
99	MINKILL	117
100	MAXFRA	118
101	INDEXNO	119
102	RADIUS	120
103	TASK	121
104	VAL	122
105	VOZ	123
106	IDHOB	124
107	TARDEFHI	125
108	TARDEFLO	126
109	MISDEF	127
110	NTINT	128
111	NAME	129
112	CCREL	133
113	PAYTBLNM	135
114	YIELD	137
115	FFRAC	138
116	PDUD	139
117	SPEEDASM	140
118	NWHDS	140
119	RELASM	141
120	RANGEASM	142

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
121	CEPASM	143
122	CEP	145
123	SPEED	146
124	NMPSIT	147
125	RANGE	148
126	ALTDLY	149
127	NALTDLY	150
128	FUNCTI	151
129	LCHINT	152
130	SIMLUN	153
131	MAXSAL	154
132	ACTIVE	155
133	IREP	157
134	PRABT	158
135	PLABT	159
136	RNGMIN	160
137	TOFMIN	161
138	CMISS	162
139	PDES	163
140	PFPF	164
141	SPDLO	157
142	RANGEDEC	158
143	RANGEREF	159
144	REL	156
145	IRECMO	160
146	IPENMO	161
147	NOINCO	188
148	TTOS	157
149	LONGLOWR	167
150	LONGUPER	168

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
151	CATCODE	169
152	MAJOR	170
153	MINOR	171
154	WACNO	172
155	BENO	173
156	IGIW	174
157	ISITE	175
158	POP	176
159	DOGLEG	177
160	ATTRLE	178
161	DISTANCE	179
162	NUMLOAD	183
163	NUMASSGN	185
164	ARRIVE	186
165	NOPERSQ	187
166	GSTART	92
167	VONBASE	189
168	NOALER	190
169	ADBLI	191
170	ADBLR	192
171	ALRTDB	193
172	ALRTDL	194
173	NADBLI	195
174	NADBLR	196
175	NLRTDB	197
176	NLRTDL	198
177	PKNAV	199
178	IREFUEL	200
179	NPRSQ1	201
180	NPRSQ2	202

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
181	NPRSQ3	203
182	NPRSQ4	204
183	NUMDBL	205
184	CAPACITY	207
185	MYRECOV1	105
186	MYRECOV2	106
187	MYRECOV3	107
188	MYRECOV4	108
189	TGTNUMB	209
190	TGTREFCD	210
191	ASIDE	211
192	NASMTYP	213
193	NBASES	214
194	NCCREG	215
195	NDCNTYCD	216
196	NDEPCRD	217
197	NPAYLOD	218
198	NPENCRD	219
199	NPNCRTY	220
200	NRECOVB	221
201	NREFUEL	222
202	NTANKERB	223
203	NTARGETS	224
204	NCOMPLX	225
205	NWEPGRP	226
206	NWEPTYP	227
207	CORBOMB#	0
208	CORMSL#	0
209	EXNBOMB#	0
210	EXNMIRV#	0

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
211	EXNMISS#	0
212	INITSTRK#	0
213	MAP#	0
214	PCTIW#	0
215	PCTPOP#	0
216	PEXBOMB#	0
217	PEXMIRV#	0
218	PEXMISS#	0
219	PFIW#	0
220	PFPOP#	0
221	PLOT#	0
222	SCALE#	0
223	SCENARIO#	0
224	LABEL#	0
225	RANGEMOD#	0
226	TARDEF#	0
227	PAYALT	137
228	NDECOYS	138
229	NAREADEC	139
230	NCMS	140
231	RETARGET#	0
232	GBASE	93
233	GPKNAV	94
234	DEFRAN	104
235	PINC	165
236	PAGELENGTH#	0
237	LINELENGTH#	0
238	REPORTCODE#	0
239	DSIDE	212
240	WEPNAME	181
241	TGTMULT	131

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D

QUICKS NON-ATTRIBUTE WORD IDENTIFICATION NUMBER

This appendix contains all dictionary non-attribute words plus their associated internal identification number and the grammatical group each word belongs to.

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
AND	4	Operator
OR	5	Operator
NOT	6	Operator
EQUAL	7	Operator
EQUALS	7	Operator
GREATER	8	Operator
LESS	9	Operator
BETWEEN	10	Operator
LIKE	11	Operator
PLUS	12	Operator
MINUS	13	Operator
TIMES	14	Operator
DIVIDED	15	Operator
POWER	16	Operator
OF	17	Operator
ALTER	1	Verb
ASSIGN	2	Verb
BUILD	3	Verb
CHANGE	4	Verb
CREATE	5	Verb
DELETE	6	Verb
DESIGN	7	Verb
EDIT	8	Verb
INDEX	9	Verb
MODIFY	10	Verb
PLANSET	11	Verb
PLOTDATA	12	Verb
PREPARE	13	Verb
PRINT	14	Verb
RESTORE	15	Verb
SAVE	16	Verb

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
SELECT	17	Verb
ASTERISK	18	Verb
ALPHAS	1	Adverb
ATTACKERS	3	Adverb
DEFENDERS	4	Adverb
DEFINE	5	Adverb
DISPLAY	6	Adverb
FIELDS	7	Adverb
FILE	8	Adverb
FIX	9	Adverb
FORMAT	10	Adverb
ONPRINTS	11	Adverb
OMITTING	12	Adverb
PLAYERS	13	Adverb
PRIORITY	14	Adverb
REPLACING	15	Adverb
SAME	16	Adverb
SETTING	17	Adverb
SORT	18	Adverb
SUPRESSING	19	Adverb
UNIT	20	Adverb
USING	21	Adverb
WHERE	22	Adverb
WITH	23	Adverb
VNOPTION	24	Adverb
KEEPING	25	Adverb
ORDER	26	Adverb
A	1	Special Word
ASCENDING	1	Special Word
AFTER	2	Special Word
D	3	Special Word

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
DESCENDING	3	Special Word
HEADER	4	Special Word
IN	5	Special Word
LINE	6	Special Word
TABLE	7	Special Word
NEW	8	Special Word
OLD	9	Special Word
OTHER	10	Special Word
PAGE	11	Special Word
REMOVE	12	Special Word
REPLACE	13	Special Word
SIDAC	14	Special Word
SPACES	15	Special Word
X	15	Special Word
TRAILER	16	Special Word
OWNED	18	Special Word
PAGENO	19	Special Word

APPENDIX E

QUICK DATA BASE PREPARATION

A QUICK data base consists of a large number of items, each item corresponding to a single target or other entity of importance to the war game. Each item consists of a number of attribute-value pairs, in which the attribute is some characteristic of the item; e.g., RADIUS, and the value gives the magnitude of the quantity. The attributes currently used in the QUICK data base are presented in appendix A, QUICK Attribute Names and Description. Not all attributes need be defined for each item. Also, the method of storing attribute values is generalized to the degree that the text English commands are.

The data base items are grouped into classes (see section 2). The allowed QUICK classes are divided into two collections: the target classes (which consist of physical installations with known coordinates) and non-target gaming classes (table 16).

Target Classes

The data record that defines a target consists of attributes as given in table 17. These attributes are needed by all target classes. Recall that any item that has a latitude and longitude and could be targeted constitutes a target record. Therefore, some of the attributes may be ignored if the target is an aim point within the game or if it is a weapon launch base. Time decay values do not have to be set for launch bases, for example.

Table 18 presents those attributes needed in addition to those outlined in table 17 if the target class is MISSIL, BOMBER, or TANKER.

If any of the developed target records are to be defined as recovery bases, the user must update that record and set attribute CAPACITY to a non-zero value via the CREATE verb.

Required attributes not set by any module, or any attribute whose default value is not appropriate must be set by the user via the CHANGE verb.

Non-target Classes

Weapon Systems. Every time a target record is created when CLASS=MISSIL, BOMBER or TANKER, the COP creates a master record for a corresponding weapon system. The master is created and the attribute TYPE is stored which equals the TYPE value for the created target class. Automatically, then, the COP provided an entry into the data base for each weapon system type. By declaring CLASS=MSLWEP (for missile weapons), BMBWEP (for bombers) or TNKWEP (for tankers) and the value for TYPE the user inputs

Table 16. Non-Target Gaming Classes

CLASS MNEMONIC	PURPOSE
WEPGRP	Contains weapon group data (formed by PLANSET)
PENCOR	Definitions of penetration and depenetration corridors
DEPCOR	
COMPLX	Contains elements of targets defined as being in a complex (formed from target classes within INDEXER)
REFUEL	Definition of refuel points
REGION	Definition of regions
PAYLOD	Identifies weapons and counts carried by a missile or bomber
BOMB	Provides warhead characteristics for gravity bombs, ASMs, single reentry vehicles (RV) and others
ASM	
RV	
MRV	
MIRV	
FACTOR	
MSLWEP	Contains weapon system characteristics for missiles, bombers and tankers
BMBWEP	
TNKWEP	
RECOV	Special entry to recovery bases

Table 17. Attributes Required for All Target Classes (Part 1 of 2)

ATTRIBUTE NAME	COMMENT
BENO	Set by JLM
CATCODE	Set by JLM
CNTRYLOC	Set by JLM
CNTRYOWN	Set by JLM
DESIG	Set by JLM
FLAG	Set by user
FVALT1	Set by user*
FVALT2	Set by user*
FVALT3	Set by user*
FVALT4	Set by user*
FVALT5	Set by user*
FVULN	Set by user
HAZ	Set by PLANSET
HAZ2	Set by PLANSET
HGZ	Set by PLANSET
HGZ2	Set by PLANSET
ICOMPL	Set by INDEXER
IDHOB	Set by user
IGIW	Set by JLM
INDEXNO	Seb by INDEXER
IREG	Seb by JLM
LAT	Set by JLM
LONG	Set by JLM
MAJOR	Set by JLM
MAXFRA	Set by user
MAXKILL	Set by user
MINKILL	Set by user
MINOR	Set by JLM
MISDEF	Set by user
NAME	Set by JLM
NHRDCOMP	Set by user
NTIMCOMP	Set by PLANSET
NTINT	Set by user
POP	Set by JLM
RADIUS	Set by JLM
SIDE	Set by JLM
TARDEFHI	Set by JLM, DBMOD or user
TARDEFLO	Set by JLM, DBMOD or user
TASK	Set by JLM
TYPE	Set by JLM
T1	Set by user*

* Set by INDEXER for class MISSIL and BOMBER.

Table 17. (Part 2 of 2)

ATTRIBUTE NAME	COMMENT
T2	Set by user*
T3	Set by user*
T4	Set by user*
T5	Set by user*
VAL	Set by user (DBMOD sets U/I targets)
VOZ	Set by PLANSET
VULN1	Set by JLM
VULN2	Set by JLM
WACNO	Set by JLM

* Set by INDEXER for class MISSIL and BOMBER.

Table 18. Attributes Required for MISSIL, BOMBER, and TANKER Target Classes

ALL

ATTRIBUTE NAME	COMMENT
ADBLI	Set by user
ADBLR	Set by user
ALRTDB	Set by DBMOD
ALRTDL	Set by user
GROUP	Set by PLANSET
IREFUEL	Set by user
ISITE	Set by user
NADBLI	Set by user
NADBLR	Set by user
NLRTDB	Set by DBMOD
NLRTDL	Set by user
NOALER	Set by DBMOD
NOINCO	Set by DBMOD
NOPERSQ	Set by DBMOD
NPRSQ1	Set by user
NPRSQ2	Set by user
NPRSQ3	Set by user
NPRSQ4	Set by user
NUMDBL	Set by user
PKNAV	Set by user
VONBASE	Set by PLANSET
WEPNAME	Set by user
BOMBER	
NOINCO	Set by DBMOD
NUMDBL	Set by user

needed attributes as outlined in table 19 . This requires a CHANGE verb and examples of exact input are:

```
CHANGE WHERE CLASS=TNKWEP & SIDE=BLUE & TYPE='KC-135'
       SETTING TTOS=3
CHANGE WHERE CLASS=MSLWEP & SIDE=BLUE &
             TYPE='MM-IA' & 'MM-IB' & 'MM-III' & 'MM-II' & TITAN
       SETTING FUNCTI=ICBM ALTDLY=0 NALTDLY=0 NMPSIT=1 SPEED=12000
          (CEP,IREP,LCHINT,PDES,PFPF,PINC,PLABT,RANGE,REL,SIMLUN)=
          ( 1,1,1,.1,.08,.85, .1,4700,.70,5) &
          (.8,2,0,.1,.07,.90, .1,5500,.75,1) &
          (.6,4,1,.1,.05,.90,.08,6300,.79,5) &
          (.6,4,0,.1,.05,.90,.08,6300,.79,1) &
          ( 1,2,0,.2,.16,.80,.17,7200,.56,1)
CHANGE WHERE CLASS=MSLWEP & SIDE=BLUE &
             TYPE='POL-A2' & 'POL-A3' & POSEID
       SETTING ALTDLY=1 FUNCTI=SLBM NALTDLY=0 NMPSIT=16
               PDES=0 SIMLUN=1
               (CEP,IREP,LCHINT,PFPF,PINC,PLABT,RANGE,REL,SPEED) =
               ( 1,4,1,.10,.86,.12,1500,.6,6000) &
               ( 1,4,0,.08,.86,.12,2500,.7,8000) &
               (.7,5,0,.10,.83,.15,2500,.8,10000)
```

WARHEAD and PAYLOD Classes. The linkage of weapon systems to payloads and warheads is very precise and the order of creation should be as outlined. Warhead characteristics are defined according to the attributes shown in table 20. Class entries are defined as:

o	BOMB	- gravity bombs
o	ASM	- air-to-surface missile
o	RV	- single shot reentry vehicle
o	MRV	- multiple reentry vehicle
o	MIRV	- multiple independent reentry vehicle
o	FACTOR	- auxiliary loading factors

Command examples that create new warhead records are:

```
CREATE SETTING CLASS=BQMB (TYPE,YIELD,PDUD,FFRAC) = ('MK-5',1.,.02,.7) &
       ('MK-7',2.,.02,.7) AND ('MK-18',4.,.02,.7) SIDE=BLUE
       SETTING CLASS = ASM (TYPE,YIELD,PDUD,FFRAC,RANGEASM,RELASM,CEPASM,
       SPEEDASM) = (HNDDOG,1.5,.02,.7,200.,.9,1.,600.) SIDE=BLUE
        SETTING CLASS= RV (TYPE,YIELD,PDUD,FFRAC) = ('MK-5',1.,.02,.7) AND
       ('MK-18',4.,.02,.7) SIDE=BLUE
       SETTING CLASS=MRV(TYPE,YIELD,PDUD,FFRAC) = ('MK-12',2.,.02,.7)
       SIDE=BLUE
       SETTING CLASS=MIRV (TYPE,YIELD,PDUD,FFRAC,NWHDS) = ('MK-17',.1,02,.7,3)
        AND ('MK-20',.05,.02,.7,10) SIDE=BLUE
CREATE SETTING CLASS=FACTOR (TYPE,NCMS,PAYALT,NDECOYS,NAREADEC)=
       (FACT1,0,HIVAL,0,0)&(FACT2,1,HIVAL,0,0)&(FACT3,1,HIGH,0,0)&
       (FACT4,2,LOW,2,0)&(FACT5,1,HIVAL,2,0)&(FACT6,0,' ',2,2)
       SIDE=BLUE
```

Table 19. Weapon Attributes (Part 1 of 2)

ATTRIBUTE NAME	COMMENT
ALL	
ACTIVE	Set by user
ALTDLY	Set by user
CEP	Set by user
FUNCTI	Set by user
LCHINT	Set by user
MAXSAL	Set by user
NALTDLY	Set by user
NMPSIT	Set by user
RANGE	Set by user
REL	Set by user
SIDE	Set by user
SIMLUN	Set by user
SPEED	Set by user
TYPE	Set by user
MSLWEP	
CMISS	Set by PLANSET from user inputs
IREP	Set by user
PDES	Set by user
PFPF	Set by user
PINC	Set by user
PLABT	Set by user
PRABT	Set by user
RNGMIN	Set by user
TOFMIN	Set by user
BMBWEP	
IPENMO	Set by user
IRECMO	Set by user
RANGEDEC	Set by user
RANGEREF	Set by user
SPDLO	Set by user

Table 19. (Part 2 of 2)

TNKWEP

ATTRIBUTE NAME	COMMENT
TTOS	Set by user

Table 20. Warhead Attributes

BOMB, ASM, RV, MRV, and MIRV

ATTRIBUTE NAME	COMMENT
FFRAC	Set by user
PDUD	Set by user
SIDE	Set by user
TYPE	Set by user
YIELD	Set by user
ASM	
CEPASM	Set by user
RANGEASM	Set by user
RELASM	Set by user
SPEEDASM	Set by user
MIRV	
NWHDS	Set by user
FACTOR	
NAREADEC	Set by user
NCMS	Set by user
NDECOYS[*]	Set by user
PAYALT	Set by user
TYPE	Set by user

[*] The range of decoys at low and high altitude is data set to 200 and 400 nautical miles respectively.

A payload table contains its name (attribute PAYTBLNM), the type(s) (attribute TYPE) of warhead(s) within the table and the number of times each warhead appears in the payload table (attribute NUMLOAD) as well as a side definition (attribute SIDE).

For proper linkage, the empty payload entries must be created first by specifying values for attribute PAYTBLNM. An example is:

```
CREATE SETTING CLASS=PAYLOD PAYTBLNM='B-47A'&'B-47B'&'B-47C'&
               B52GH1&B52GH2&'B-52E1'&'B-52E2'&'B-58'&
               'MM-II'&'MM-IB'&'MM-IA'&'TITAN'&
               POSEID&'POL-A2'&'POL-A3' SIDE=BLUE
```

Now the user links each payload table to its warheads. Examples are:

```
CREATE SETTING CLASS=BOMB (PAYTBLNM,TYPE,NUMLOAD)=
       ('B-47A','MK-5',2)&('B-47B','MK-7',2)&
       ('B-47C','MK-18',1)&(B52GH1,'MK-5',4)&
       (B52GH2,'MK-18',2)&('B-52E1','MK-7',2)&
       ('B-52E2','MK-7',2)&('B-58','MK-18',1)&
       ('B-58','MK-5',3) SIDE=BLUE
CREATE SETTING CLASS=ASM TYPE=HNDDOG NUMLOAD=2
               SIDE=BLUE PAYTBLNM=B52GH1&B52GH2&'B-52E2'
       SETTING CLASS=RV SIDE=BLUE (PAYTBLNM,TYPE,NUMLOAD)=
               ('MM-IB','MK-5',1)&('MM-IA','MK-5',1)&
               (TITAN,'MK-18',1)
```

Weapon Base to Payload Link. Linkage must now be established between the created payload tables and the weapon bases that refer to these tables for definition concerning delivery systems. The immediate thought is a connection between PAYTBLNM name and the value for the weapon base attribute TYPE. While in most cases this is the connection, it is too restrictive for general use. The user does not have the flexibility to identify individual bases as being separate from the normal similar types. Therefore, a weapon subtype is defined whereby linkage is created between payload tables and individual weapon bases. The connection is through attribute WEPNAME which is first linked to payloads and then defined for individual bases. Commands that link WEPNAME to payload tables could be:

```
CREATE SETTING CLASS=BMBWEP SIDE=BLUE
                  (WEPNAME,TYPE,PAYTBLNM)=
              ('B-52H1','B-52H' ,B52GH1 ) &
              ('B-52H2','B-52H' ,B52GH2 ) &
              ('B-52G1','B-52G' ,B52GH1 ) &
              ('B-52G2','B-52G' ,B52GH2 ) &
              ('B-52E1','B-52E' ,'B-52E1') &
              ('B-52E2','B-52E' ,'B-52E2') &
              ('B-53A' ,'B-58'  ,'B-58'  ) &
              ('F-111A','F-111' ,'B-47A' ) &
              ('F111BA','F-111B','B-47A')
CREATE SETTING CLASS=MSLWEP SIDE=BLUE
              (WEPNAME,TYPE,PAYTBLNM)=
              ('MM-IAA','MM-IA','MM-IA') &
              ('MM-IBA','MM-IB','MM-IB') &
              (MMIIIA,'MM-III','MM-II') &
              ('MM-IIA','MM-II','MM-II') &
              (TITANA,TITAN,TITAN) &
              (POLA2A,'POL-A2','POL-A2') &
              (POLA3A,'POL-A3','POL-A3') &
              (POSIDA,POSEID,POSEID)
```

The final step of weapon linkage is to relink the weapon bases to the new weapon subtype (WEPNAME). Originally, the COP linked all bases to a dummy subtype and, therefore, relinkage is accomplished through a CHANGE verb and a WHERE clause used to identify the subset of bases in question and the SETTING *clause used to name the subtype* (WEPNAME) into which these bases fall. Examples are:

```
CHANGE WHERE CLASS=BOMBER AND SIDE=BLUE
       AND CATCODE BETWEEN 95 AND 98
       SETTING WEPNAME='B-52G2'
CHANGE WHERE CLASS=BOMBER AND SIDE=BLUE AND TYPE='B-52E'
       SETTING WEPNAME='B-52E1'
CHANGE WHERE CLASS=BOMBER AND SIDE=BLUE
       AND CATCODE BETWEEN 103 AND 111
       SETTING WEPNAME='B-52E2'
```

Geographic Classes. Penetration/depenetration corridor (CLASS=PENCOR or DEPCOR) and refuel points (CLASS=REFUEL) may be created at any stage prior to the execution of the PREPALOC module. Required attributes are given in table 21.

When creating corridors, the user should create the corridor itself first and in a separate verb, create the doglegs of the corridor. An example of a command that creates many penetration corridors is:

Table 21. Geographic Attributes

PENCOR

ATTRIBUTE NAME	COMMENT
ATTRCO	Set by user
ATTRLE	Set by user (dogleg attribute)
ATTRSU	Set by user
CORNUM	Set by user (corridor number)
DEFRAN	Set by user
DOGLEG	Set by user (dogleg attribute)
HILOAT	Set by user
KORSTY	Set by user
LAT	Set by user (dogleg attribute)
LONG	Set by user (dogleg attribute)
ORLAT	Set by user
ORLONG	Set by user

DEPCOR

CORNUM	Set by user (corridor number)
DOGLEG	Set by user (dogleg attribute)
LAT	Set by user (dogleg attribute)
LONG	Set by user (dogleg attribute)
MYRECOV1	Set by user (base DESIG)
MYRECOV2	Set by user (base DESIG)
MYRECOV3	Set by user (base DESIG)
MYRECOV4	Set by user (base DESIG)

REFUEL

IREG	Set by user
LAT	Set by user
LONG	Set by user

```
CREATE SETTING CLASS=PENCOR SIDE=BLUE
       (CORNUM,ORLAT,ORLONG,KORSTY,HILOAT,DEFRAN,ATTRSU,ATTRCO)=
       (  1, 50,350,1. .2,250,.0003,   .001) &
       (  2, 50,350,1. .2,250,.0003,   .001) &
       (  3, 55,215,4. .2,250,.0001, .0003) &
       (  4, 60,230,4. .2,250,.0001, .0003) &
       (  5, 50,235,4. .2,250,.0001, .0003) &
       (  6, 40,243,4. .2,250,.0001, .0003) &
       (  7, 60,275,2. .2,250,0.    , .0006)
```

For each corridor, doglegs start at the corridor origin and work outward. A line segment is identified through attribute DOGLEG. Each corridor will store line segments sorted on the attribute DOGLEG. Any number is permitted for DOGLEG entries. It is suggested that DOGLEG values be initially entered in multiples of 10 (10, 20, 30, etc.). This numbering will permit ease of redefinition if added line segments are defined after initial creation. A command that links DOGLEGs to corridors is:

```
CREATE SETTING CLASS=PENCOR SIDE=BLUE DOGLEG=10
       (CORNUM,LAT,LONG,ATTRLE)=
       ( 3,62,172,0.    ) & ( 4,68,193,0.    ) &
       ( 5,40,227,0.    ) & ( 6,36,236,0.    ) &
       ( 7,73,275,0.    ) & ( 8,52.5,349,.0005) &
       ( 9,43,342,0.    ) & (10,45,325,0.    ) &
       (11,71,330,0.    ) & (12,60,340,0.    )
```

Depenetration corridor creation is similar as for penetration corridors except that links must be established between corridor exit and permissable recovery bases. Attributes MYRECOV1, MYRECOV2, MYRECOV3, and MYRECOV4 define the DESIG of the recovery bases where an aircraft may land if exiting a given corridor. No other action is required for linkage. Commands are:

```
CREATE SETTING CLASS=DEPCOR SIDE=BLUE
       (CORNUM,MYRECOV1,MYRECOV2,MYRECOV3,MYRECOV4)=
       (1,AB800,AB801.AB802,AB803) &
       (2,AB804,AB805.AB806,AB807) &
       (3,AB500,AB501.AB502,AB503) &
       (4,AB504,'0','0','0') &
       (5,AB505,AB506.'0','0')

 CREATE SETTING CLASS=DEPCOR SIDE=BLUE
        (CORNUM,DOGLEG.LAT,LONG)=
        (1,1,43,222) & (1,2,43,222) &
        (2,1,38,227) & (2,2,38,227) &
        (3,1,42,331) & (3,2,42,331) &
        (4,1,42,320) & (4.2,42,320) &
        (5,1,37,305) & (5,2,37,305) &
        (6,1,37,288) & (6,2,37,288)
```

Refuel points are created through a simple definition of latitudes and longitudes.

APPENDIX F

EXECUTION OF THE QUICK SYSTEM

This appendix presents the detailed characteristics of user procedures for executing the QUICK system or more appropriately the execution of QUICKs executive software, the COP. The capability enables the user to submit batch-mode jobs or to execute remote terminal jobs.

These capabilities and the necessary data file definition are further described in the following subsections.

BATCH

The COP is executed from a system loadable H* file. As a result, the user is not required to specify within the Job Control Language (JCL) which modules will be executed. COP will determine which modules are required from the input text English command sentences. A generalized set of JCL which will execute the COP appears in figure 25. The limits given may be altered according to the user's expected printed output and run time. The various tape and file units requested need not all be included as some are not utilized by all modules. Table 22 contains a list of the tape and file units and which ones may be omitted depending upon which modules will be executed.

Remote Terminal Entry

A time sharing capability exists whereby a user may execute the COP or make source code corrections to a COP module. Procedures to utilize this function are outlined below. In the following narrative, commands to be entered by the user will be flagged by an arrowhead (➤).

Step 1. Log On and Initiation. The first step involved is to log on to the HIS system and to specify what operating system is to be used and, then, what modules are to be executed. Steps involved in this example utilizes a catalog file called 631IDP00/COPRUN which contains the procedures to execute the COP. These steps are:

```
➤ $*$LOG24,TSS
  TERMINAL CK
  USERID$PASSWORD
➤       (enter userid and password)
  IDENT?
➤       (enter ident card information)
  0  BLOCKS FILE SPACE AVAILABLE
  CLASSIFICATION OF YOUR OUTPUT
➤       (enter classification code, i.e., UZZ)
  CLASSIFICATION OF FILES YOU WILL CREATE
➤       (enter classification code, i.e., UZZ)
```

```
$       IDENT    5162,COPRN,314,I M A USER,631,12
$       USERID   631IDP00$PASSWORD/VHC
$       LIMITS   30,49K,,20K
$       PROGRAM  RLHS,DUMP
$       LIMITS   30,49K,,20K
$       PRMFL    H*,R/W,R,631IDP00/TEST/COP/HSTAR
$       PRMFL    QD,R/W,R,631IDP00/TEST/COP/IDS
$       FILE     02,X02R,100L
$       FILE     08,X08R,100L
$       FILE     19,X19R,100L
$       TAPE9    20,X20D,,12345,,INPUT-JAD
$       FILE     21,X21R,100L
$       FILE     22,X22R,100L
$       FILE     23,X23R,100L
$       FILE     24,X24R,100L
$       FILE     25,X25R,100R
$       FILE     30,X30R,10L
$       TAPE9    31,X31D,,54321,,OUTPUT-SPILLTAPE
$       TAPE9    35,X35D,,67890,,OUTPUT-TAPE1
$       TAPE9    36,X36D,,98765,,OUTPUT-TAPE2
$       DATA     I*
    :         :         :
    :         :         :
  Text  English  Command  Sentences
    :         :         :
    :         :         :
$       ENDJOB
***EOF
```

Figure 26. Batch-mode JCL

Table 22. Batch-mode JCL File Utilization

Unit File Code	Comment
H*	Must be included; contains system loadable COP modules.
QD	Must be included; contains the COP IDS data base file
02	TGTFIL used by IIM
08	BASFIL used by IIM
19	TARFIL used by IIM
20	JAD input unit. Used by JLM
21 22 23 24	Internal sort files. Used by: JLM, REPORT, EIM, INDEXER, PLANSET
25	Random storage file used in concert with sort: JLM, REPORT, EIM, INDEXER
30	Directory file, used by IIM
31	*Spill tape, used by IIM*
35 36	Output tape files, used by EIM and JLM

```
    SYSTEM ?
►   YFORT O 631IDPOO/COPRUN
    READY
    *
►   RUN COPRUN
```

The user now is interacting with a YFORT TSS subsystem program. The YFORT program will signal its desire for a response by outputting an equal sign (=) to which the user responds accordingly.

After the system outputs an equal sign, the user has multiple responses in some cases. In order to assist the user, a response of HELP may be entered and the system will respond with all the possible replies. Permissible use of the HELP command is outlined below.

Step 2. Mode Selection. The YFORT program is asking if the user desires to execute the COP, update the source, or both by displaying the command:

```
    ENTER MODE (RUN, COMPILE)
    =
```

If a run only is desired enter RUN and skip to step 4; else enter

```
►   COMPILE
```

Step 3. Compiling an Updated COP. The user is compiling module(s) and the program asks which one(s) by displaying:

```
    WHICH MODULES (SEPARATE BY COMMAS)
         (HELP WILL LIST THE MODULES)
```

A possible response would be:

```
►   DATA, JLM
```

In order to ascertain if this is a compilation and execution, the program displays:

```
    ENTER RUN MODE?  (YES, NO)
    =
```

If the user wishes to compile only, enter NO and skip to step 6; otherwise enter YES.

Step 4. Limit Specification. The COP executes a wide variety of modules and differing size data bases which implies varying limits as to computer resource usage in terms of computer time, core storage, and lines to be printed. A series of commands are given to provide interactive capability whereby the limits may be altered. First, the system must know which modules (or verbs) are to execute. Hence the command:

```
WHICH VERBS (SEPARATE BY COMMAS)
     (HELP WILL LIST THE VERBS)
=
```

asks what is to be executed by this submittal. A possible response is:

```
➤ CREATE, CHANGE, PRINT
```

The program responds with a set of default limits such as

```
LIMITS ARE 30,49K,,20K
NEW LIMITS? (ENTER NO OR ENTER NEW LIMITS)
=
```

If the displayed limits are valid, a NO response is entered. Otherwise new limits are entered. For example:

```
➤ 20,49K,,20K
```

Step 5. Text English Command Entry. User inputs are now given. The display command is:

```
INPUT DATA FILE (ENTER CATALOG FILE STRING OR DONE)
=
```

Here the program is asking the user to name a data file in which the user has stored text English commands. The user will be able to enter any number of such files, in the order as input. The user will also be able to enter text English commands directly and interweave files with direct entry by following this procedure. If the user enters the catalog file string of a file, the program will return to the beginning of step 5. If the user has entered all the files names desired, DONE is entered.

The program now requests the possibility of text English commands entered directly by displaying:

```
ENTER DATA FROM TERMINAL (BLANK USE TERMINATES)
=
```

All inputs are entered followed by a blank line.

To allow for the possibility of additional data, the program displays:

```
ANY ADDITIONAL DATA (YES,NO)
```

If YES is entered, the program returns to the beginning of step 5. A NO response terminates step 5.

Step 6. Executing the COP. The constructed job may now be submitted which must be done within the CARDIN subsystem of HIS. The following is displayed:

```
    THANK YOU
                    RUN
    THE JOB TO            COP HAS BEEN BUILT
                  COMPILE
    TO EXECUTE ENTER -
    SYST CARD N
    THEN -
    RUN THE JOB

➤   SYST CARD N
    READY
    *
➤   RUN THE JOB

    SNUMB # XXXXX
    *
```

The job has now been entered, the user should note the SNUMB.

Preparing an IDS File for COP

The COP must run on an IDS file. When the user wishes to restore onto a previously unused file or build a QUICK data base from scratch, he must use a file specially prepared for IDS. The creation of such a file has two steps. First, the user must create the file using the File Management Subsystem (FMS). Besides the usual options employed to create a random file, the following additional options are used:

BASESIZE/N/ - N defines the maximum number of pages in the IDS data file

RNG/r1,r2/ - Defines the page range (for QUICK applications set r1=1 and r2=N as above)

LINESPERPAGE/m/ - m defines the number of lines on an IDS page (for QUICK applications set m=21)

An example of an appropriate FCREATE directive would be:

```
FCREATE/IDS/ MYFILE, BASESIZE/401/,RNG/1,401/,LINESPERPAGE/21/,
              SIZE/102/,MODE/RNAD/,FCLASS/UZZ/
```

Following the creation of the IDS file, the user must initialize the file by utilizing an IDS utility called QUTI. This utility has a single input card that specifies the page range to be initialized. An example of an appropriate QUTI activity would be:

```
$     PROGRAM  QUTI
$     PRMFL    A1,R/W,R,631IDPOO/MYFILE
IDS   INITIAL  1,401
```

If the user needs any further assistance, consult Honeywell Reference DC53A, Rev.0 #I-D-S/I USER'S GUIDE.

IDS File Unlock

If a run of the QUICK system aborts during processing, very often the user will be informed that the IDS file is abort locked. This message will appear in the print report. This condition must be corrected before any further processing can occur. Therefore, the user must 'unlock' the IDS data file. This is done via a FILSYS activity which contains a directive similar to the following:

```
ALOCK    631IDPOO/TEST/COP/IDS,OFF
```

THIS PAGE INTENTIONALLY LEFT BLANK

GLOSSARY

Circular Error Probable (CEP): An indicator of the delivery accuracy of a weapon system, used as a factor in determining probable damage to a target. It is the radius of a circle within which half of the missiles/ projectiles are expected to fall.

Computer Program: A program expressed in computer code designed to solve a class of problems, or specializing on a specific problem when appropriate parametric values are supplied.

Computerized War Gaming Model: A computer program, or series of programs, designed to simulate the logic of actions or interactions of a conflict situation and provide results for subsequent analysis.

Cross Targeting: The deliberate mixing of weapon types assigned to individual targets so that unexpectedly high failure rates for any specific weapon type, or group of weapons, will not result in a catastrophic failure of the war plan.

Damage Expectancy (DE): Probability of achieving a desired level of damage considering the probability of weapon arrival (PA) and the probability of damage (PD), i.e., DE=PAxPD.

Data Base: An organized collection of data records with similar or associated characteristics either to be operated upon by a system or contributing to the operation of a system.

Directing Assignment: A term used within QUICK to identify weapon-to-target assignments which are directed (fixed) by the user.

Dynamic: Pertaining to a quantity which is affected by time, energy, or power, and therefore indicates a relatively transient or unstable (changing) condition.

ESP Model: The Event Sequenced Program used by the Joint Strategic Target Planning Staff (JSTPS) to simulate large-scale strategic warfare.

Event: A happening in time, either within a simulation or in reality.

Expected Value: The average or mean value which would be obtained if a given event were repeated many times.

Fixed Assignments (weapon fixes): A term used within QUICK to identify weapon-to-target assignments which are directed (fixed) by the user.

FLAG: A code used in imposing weapon group restrictions within QUICK

General War: Armed conflict between major powers in which the total resources of the belligerents are employed, and the national survival of a major belligerent is in jeopardy.

Input: Any factors, data, parameters, values, or instructions required for proper operation of a model or submodel to produce game results.

Limited War: Armed conflict short of general war, exclusive of incidents, involving the overt engagement of the military forces of two or more nations.

NEMO Model: The Nuclear Exchange MOdel maintained by the Navy for simulation of two-sided global nuclear war.

Nuclear Vulnerability Assessment: The estimation of probable or expected effects of hypothetical nuclear attacks on population, forces, and resources.

Optimizing: Selecting the most advantageous solution or course of action in a competitive situation within constraints imposed by the conditions of the problem.

Posture: Relative place or position; state or condition at a given time, especially in relation to other persons or things.

Probability of Arrival (PA): The probability of a delivery vehicle delivering a weapon which detonates as planned.

Probability of Damage (PD): The probability that damage will occur to a target expressed as a percentage or as a decimal.

RISOP: A hypothetical Red Integrated Strategic Offensive Plan.

SIDAC: Single Integrated Damage Analysis Offensive Plan.

SIOP: The Single Integrated Operational Plan.

Terminal ABM Defense: The ballistic missile defense provided for a single target or set of collocated targets by short-range defensive systems. QUICK uses a subtractive terminal defense model. Each incoming warhead or terminal decoy is assigned an interceptor until the local supply of defensive missiles is exhausted.

DISTRIBUTION

Addressee	Copies
CCTC Codes	
Technical Library (C124)	3
C124 (Stock)	6
C313	1
C314	11
C400	1
C600	1
C730	1
DCA Code	
205	1
EXTERNAL	
Chief, Studies, Analysis and Gaming Agency, OJCS ATTN: SFD, Room 1D957, Pentagon, Washington, DC 20301	5
Chief of Naval Operations, ATTN: OP-96C4, Room 4A478, Pentagon, Washington, DC 20350	2
Commander-in-Chief, North American Air Defense Command ATTN: NPXYA, Ent Air Force Base, CO 80912	2
Commander, U. S. Air Force Weapons Laboratory (AFSC) ATTN: AFWL/SAB, Kirtland Air Force Base, NM 87117	1
Commander, U. S. Air Force Weapons Laboratory (AFSC) ATTN: AFWL/SUL (Technical Library), Kirtland Air Force Base, NM 87117	1
Director, Strategic Target Planning, ATTN: (JPS), Offutt Air Force Base, NE 68113	2
Defense Documentation Center, Cameron Station, Alexandria, VA 22314	12
	50

THIS PAGE INTENTIONALLY LEFT BLANK

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
CSM UM 9-[redacted], Volume I
77

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK), Users Manual, Data Management Subsystem

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Dale J. Sanders
Paul F. M. Maykrantz
Jim M. Herrin
Edward F. Bersson

8. CONTRACT OR GRANT NUMBER(s)
✓ DCA 100-75-C-0019

9. PERFORMING ORGANIZATION NAME AND ADDRESS
✓
System Sciences, Incorporated
4720 Montgomery Lane
Bethesda, Maryland 20014

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Command and Control Technical Center
Room BE-685, The Pentagon,
Washington, DC 20301

12. REPORT DATE
1 June 1977

13. NUMBER OF PAGES
170

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
War Gaming, Resource Allocation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide statistical output summaries, and produce input tapes to simulator subsystems external to QUICK.

The Users Manual consists of four volumes which are intended for the user/analyst who is concerned with preparing the data base for a war game, selecting optional features of QUICK, designating control parameters, submitting computer jobs, and analyzing computer output. This volume, Volume I, provides detailed

20. ABSTRACT (Continued).

instructions for execution of the Data Management Subsystem and the modules it comprises.

The Users Manual complements the other QUICK Manuals to facilitate application of the war gaming system. These manuals (Series 9-77 for Volumes I & II, Series 9-74 for Volumes III & IV) are published by the Command and Control Technical Center (CCTC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.